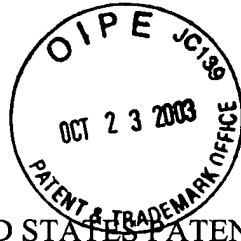


00862.023209



PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	:	Examiner: Not Yet Assigned
ATSUSHI NAKAMURA	)	
	:	Group Art Unit: NYA
Application No.: 10/653,962	)	
	:	
Filed: September 4, 2003	)	
	:	
For: INFORMATION PROCESSING	)	
SYSTEM, INFORMATION	:	
PROCESSING APPARATUS,	)	
AND INFORMATION	:	
PROCESSING METHOD	)	October 22, 2003

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

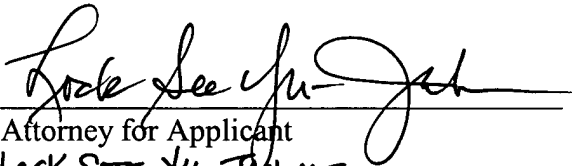
Sir:

In support of Applicant's claim for priority under 35 U.S.C. § 119, enclosed is  
a certified copy of the following Japanese application:

2002-260396, filed September 5, 2002.

Applicant's undersigned attorney may be reached in our New York office by telephone at (212) 218-2100. All correspondence should continue to be directed to our address given below.

Respectfully submitted,

  
Attorney for Applicant  
LOCK SEE YU-JUANES  
Registration No. 38,667

FITZPATRICK, CELLA, HARPER & SCINTO  
30 Rockefeller Plaza  
New York, New York 10112-3800  
Facsimile: (212) 218-2200

NY\_MAIN 383667v1

日本国特許庁  
JAPAN PATENT OFFICE

CF 1703209  
U.S. Appl. No. 10/653,962  
US

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2002年 9月 5日  
Date of Application:

出願番号 特願2002-260396  
Application Number:  
[ST. 10/C]: [JP 2002-260396]

出願人 キヤノン株式会社  
Applicant(s):

2003年 9月24日

特許庁長官  
Commissioner,  
Japan Patent Office

今井康夫



【書類名】 特許願

【整理番号】 4578019

【提出日】 平成14年 9月 5日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/00

【発明の名称】 情報処理システム、情報処理装置、情報処理方法、プログラム及び記憶媒体

【請求項の数】 11

【発明者】

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会社  
社内

【氏名】 中村 敦

【特許出願人】

【識別番号】 000001007

【氏名又は名称】 キヤノン株式会社

【代理人】

【識別番号】 100076428

【弁理士】

【氏名又は名称】 大塚 康德

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100112508

【弁理士】

【氏名又は名称】 高柳 司郎

【電話番号】 03-5276-3241

## 【選任した代理人】

【識別番号】 100115071

【弁理士】

【氏名又は名称】 大塚 康弘

【電話番号】 03-5276-3241

## 【選任した代理人】

【識別番号】 100116894

【弁理士】

【氏名又は名称】 木村 秀二

【電話番号】 03-5276-3241

## 【手数料の表示】

【予納台帳番号】 003458

【納付金額】 21,000円

## 【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0102485

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理システム、情報処理装置、情報処理方法、プログラム及び記憶媒体

【特許請求の範囲】

【請求項 1】

通信制御バスで接続された第 1、第 2 機器を含む情報処理システムであって、  
前記第 1 機器は、ロジカルユニット 1 つを使ってひとつのコマンド O R B に対して複数割り当てられたデータバッファを有し、

前記第 2 機器は、複数の前記データバッファのいずれか一つに対するデータ通信が完了したことを通知する完了通知手段を有し、

前記第 1 機器は、更に、前記完了通知手段による通知に基づき、データ通信が完了していないデータバッファに対しては更新を行わず、完了しているデータバッファに対して更新を行う手段を有することを特徴とする情報処理システム。

【請求項 2】

前記通信制御バスは、I E E E 1 3 9 4 に準拠した通信制御バスであり、前記第 1 機器と前記第 2 機器との間では、シリアルバスプロトコル 3 を通信プロトコルとしてデータ通信を行うことを特徴とする請求項 1 記載の情報処理システム。

【請求項 3】

前記完了通知手段は、前記第 1 機器のステータス F I F O に、前記 2 つのデータバッファの内、データ通信が完了したデータバッファを示すステータスを発行することを特徴とする請求項 1 または 2 に記載の情報処理システム。

【請求項 4】

前記第 1 機器は、前記複数のデータバッファに格納されているデータの識別情報を前記 O R B に含むことを特徴とする請求項 1、2 または 3 に記載の情報処理システム。

【請求項 5】

通信制御バスで接続された 2 つの機器間における通信方法であって、

1 つのコマンド O R B に対し複数のデータバッファを参照する工程と、

前記 O R B で指定される複数のデータバッファのいずれかがデータ通信を完了

した場合に、その完了を通知する完了通知工程と、  
前記データバッファの更新情報を通知する工程と、  
を含むことを特徴とする通信方法。

【請求項 6】

他の機器と通信制御バスで接続された情報処理装置であって、  
ロジカルユニット 1 つを使ってひとつのコマンド O R B に対して複数割り当て  
られたデータバッファと、

前記他の機器から、複数の前記データバッファのいずれか一つに対するデータ  
通信が完了したことを示す完了通知を受信する手段と、

前記完了通知に基づき、データ通信が完了していないデータバッファに対して  
は更新を行わず、完了しているデータバッファに対して更新を行う手段とを有す  
ることを特徴とする情報処理装置。

【請求項 7】

他の機器と通信制御バスで接続された情報処理装置であって、

ロジカルユニット 1 つを使って、ひとつのコマンド O R B に対し複数のデータ  
バッファを用意した他の機器に対し、データ通信を完了していないデータバッ  
ファに対しては更新を行わず、完了しているデータバッファに対して更新を行わ  
せるために、前記複数のデータバッファのいずれか一つに対するデータ通信が完  
了したことを通知する完了通知手段を有することを特徴とする情報処理装置。

【請求項 8】

他の機器と通信制御バスで接続された情報処理装置における通信方法であって

、  
ロジカルユニット 1 つを使って、ひとつのコマンド O R B に対し複数のデータ  
バッファを用意し、

前記他の機器から、前記複数のデータバッファのいずれか一つに対するデータ  
通信が完了したことを示す完了通知を受信し、

前記完了通知に基づき、完了していないデータバッファに対して更新を行わず  
、完了しているデータバッファに対して更新を行うことを特徴とする通信方法。

【請求項 9】

他の機器と通信制御バスで接続された情報処理装置における通信方法であって

、

ロジカルユニット 1 つを使って、ひとつのコマンド O R B に対し複数のデータバッファを用意した他の機器に対し、完了していないデータバッファに対して更新を行わず、完了しているデータバッファに対して更新を行わせるために、前記複数のデータバッファのいずれか一つに対するデータ通信が完了したことを通知することを特徴とする通信方法。

#### 【請求項 1 0】

請求項 5， 8 または 9 に記載の通信方法をコンピュータに実現させることを特徴とするプログラム。

#### 【請求項 1 1】

前記請求項 1 0 に記載のプログラムを格納したことを特徴とするコンピュータ読取り可能な記憶媒体。

#### 【発明の詳細な説明】

##### 【 0 0 0 1】

##### 【発明の属する技術分野】

本発明は、I E E E 1 3 9 4 等のインターフェースで接続される情報処理システム、情報処理装置、情報処理方法、プログラム及び記憶媒体に関するものである。

##### 【 0 0 0 2】

##### 【従来の技術】

セントロニクス、U S B、そして I E E E 1 3 9 4 等の各種インターフェースにおいてホストとデバイスの 1 対 1 接続、通信を行なう場合、ホスト・デバイス間でデバイスの各種機能を制御するためにために通信プロトコルの 1 セッション中に複数の論理チャネル接続を樹立し、チャネルごとにデータ転送を行なう。

##### 【 0 0 0 3】

I E E E 1 3 9 4 上のデータ通信プロトコルである S B P (シリアルバスプロトコル) - 2 は、それ自体では、ひとつのロジカルユニットにおいて、データ転送の単位である O R B (オペレーションリクエストブロック) 1 個に対して 1 つの



データバッファが参照されるため、片方向の単一データチャネル、片方向半二重のデータ通信を行なう仕組みになっており、複数論理チャネルを実現するのは困難であった。現在規格策定中の S B P - 2 の機能拡張版である S B P - 3 においては、この点に注目し、L U N (ロジカルユニット) におけるデータ転送の単位である O R B に対して 2 つのデータバッファを参照できるようにして 1 L U N につき 2 つのデータ転送チャネルを実現することを可能とする拡張が行なわれた。

#### 【 0 0 0 4 】

##### 【発明が解決しようとする課題】

しかしながら S B P - 3 のデータ転送においては 2 つのデータ転送チャネルのフローコントロールはあくまでも O R B 単位で行なわれるため、何らかの理由によりどちらかのデータ転送チャネルにおいてデータの送受信ができなくなった場合、またどちらかのデータ転送チャネルのデータ転送が他のチャネルよりも遅い場合、正常に転送されているチャネル、速いほうのチャネルが影響を受けてしまう。

#### 【 0 0 0 5 】

これは、片方のデータバッファに対するアクセスが正常終了しても、もう片方のデータバッファアクセスが完了しないとその O R B に対する完了通知を S B P - 3 のイニシエータに対して通知できず、L U N としてのデータ転送が滞ってしまうためである。

#### 【 0 0 0 6 】

本発明は、上記従来技術の課題を解決するためになされたもので、その目的とするところは、I E E E 1 3 9 4 バス等の 2 つのチャネルを用いて効率的なデータ転送を行うことにある。

#### 【 0 0 0 7 】

##### 【課題を解決するための手段】

上記目的を達成するため、本発明に係るシステムは、  
通信制御バスで接続された第 1、第 2 機器を含む情報処理システムであって、  
前記第 1 機器は、ロジカルユニット 1 つを使ってひとつのコマンド O R B に対して複数割り当てられたデータバッファを有し、

前記第 2 機器は、複数の前記データバッファのいずれか一つに対するデータ通信が完了したことを通知する完了通知手段を有し、

前記第 1 機器は、更に、前記完了通知手段による通知に基づき、データ通信が完了していないデータバッファに対しては更新を行わず、完了しているデータバッファに対して更新を行う手段を有することを特徴とする。

#### 【0 0 0 8】

前記通信制御バスは、I E E E 1 3 9 4 に準拠した通信制御バスであり、前記第 1 機器と前記第 2 機器との間では、シリアルバスプロトコル 3 を通信プロトコルとしてデータ通信を行うことを特徴とする。

#### 【0 0 0 9】

前記完了通知手段は、前記第 1 機器のステータス F I F O に、前記 2 つのデータバッファの内、データ通信が完了したデータバッファを示すステータスを発行することを特徴とする。

#### 【0 0 1 0】

前記第 1 機器は、前記複数のデータバッファに格納されているデータの識別情報を前記 O R B に含むことを特徴とする。

#### 【0 0 1 1】

上記目的を達成するため、本発明に係る方法は、  
通信制御バスで接続された 2 つの機器間における通信方法であって、  
1 つのコマンド O R B に対し複数のデータバッファを参照する工程と、  
前記 O R B で指定される複数のデータバッファのいずれかがデータ通信を完了した場合に、その完了を通知する完了通知工程と、  
前記データバッファの更新情報を通知する工程と、  
を含むことを特徴とする。

#### 【0 0 1 2】

上記目的を達成するため、本発明に係る装置は、  
他の機器と通信制御バスで接続された情報処理装置であって、  
ロジカルユニット 1 つを使ってひとつのコマンド O R B に対して複数割り当てられたデータバッファと、

前記他の機器から、複数の前記データバッファのいずれか一つに対するデータ通信が完了したことを示す完了通知を受信する手段と、

前記完了通知に基づき、データ通信が完了していないデータバッファに対しては更新を行わず、完了しているデータバッファに対して更新を行う手段とを有することを特徴とする。

#### 【0013】

他の機器と通信制御バスで接続された情報処理装置であって、

ロジカルユニット1つを使って、ひとつのコマンドORBに対し複数のデータバッファを用意した他の機器に対し、データ通信を完了していないデータバッファに対しては更新を行わず、完了しているデータバッファに対して更新を行わせるために、前記複数のデータバッファのいずれか一つに対するデータ通信が完了したことを通知する完了通知手段を有することを特徴とする。

#### 【0014】

他の機器と通信制御バスで接続された情報処理装置における通信方法であって、

ロジカルユニット1つを使って、ひとつのコマンドORBに対し複数のデータバッファを用意し、

前記他の機器から、前記複数のデータバッファのいずれか一つに対するデータ通信が完了したことを示す完了通知を受信し、

前記完了通知に基づき、完了していないデータバッファに対して更新を行わず、完了しているデータバッファに対して更新を行うことを特徴とする。

#### 【0015】

他の機器と通信制御バスで接続された情報処理装置における通信方法であって、

ロジカルユニット1つを使って、ひとつのコマンドORBに対し複数のデータバッファを用意した他の機器に対し、完了していないデータバッファに対して更新を行わず、完了しているデータバッファに対して更新を行わせるために、前記複数のデータバッファのいずれか一つに対するデータ通信が完了したことを通知することを特徴とする。

**【0016】**

上記目的を達成するため、本発明に係るプログラムは、上記通信方法をコンピュータに実現させる。

**【0017】**

上記目的を達成するため、本発明に係る記憶媒体は、上記プログラムを格納する。

**【0018】****【発明の実施の形態】**

以下に、図面を参照して、この発明の好適な実施の形態を例示的に詳しく説明する。ただし、この実施の形態に記載されている構成要素の相対配置、表示画面等は、特に特定の記載がない限りは、この発明の範囲をそれらのみに限定する趣旨のものではない。

**【0019】****(第1実施形態)**

本発明の第1実施形態としての情報処理システムについて説明する前に、IEEE 1394の技術の概要について説明する。

**【0020】****〈IEEE 1394の技術の概要〉**

以下、本実施の形態のデジタルインタフェースに適用されるIEEE 1394-1995規格の技術について簡単に説明する。尚、IEEE 1394-1995規格（以下、IEEE 1394規格）についての詳細は、1996年の8月30日にIEEE（The Institute of Electrical and Electronics Engineers, Inc.）から出版された「IEEE Standard for a High Performance Serial Bus」に記述されている。

**【0021】****(1) 概要**

図2にIEEE 1394規格に準拠したデジタルインタフェース（以下、1394インタフェース）を具備するノード（機器）により構成される通信システム（以下、1394ネットワーク）の一例を示す。1394ネットワークは、シリ

アルデータの通信可能なバス型ネットワークを構成するものである。

#### 【 0 0 2 2 】

図 2 において、各ノード A ~ F は、I E E E 1 3 9 4 規格に準拠した通信ケーブルを介して接続されている。これらのノード A ~ H は、例えば、P C (Personal Computer)、デジタル V T R (Video Tape Recorder)、D V D (Digital Video Disc) プレーヤ、デジタルカメラ、ハードディスク、モニタ等の電子機器である。

#### 【 0 0 2 3 】

1 3 9 4 ネットワークの接続方式は、ディジーチェーン方式とノード分岐方式とに対応しており、自由度の高い接続を可能としている。

#### 【 0 0 2 4 】

又、1 3 9 4 ネットワークでは、例えば、既存の機器を削除したり、新たな機器を追加したり、既存の機器の電源を O N / O F F したりした場合に、自動的にバスリセットを行う。このバスリセットを行うことにより、1 3 9 4 ネットワークは、新たな接続構成の認識と各機器に対する I D 情報の割り当てとを自動的に行うことができる。この機能によって、1 3 9 4 ネットワークは、ネットワークの接続構成を常時認識することができる。

#### 【 0 0 2 5 】

又、1 3 9 4 ネットワークは、他の機器から転送されたデータを中継する機能を有している。この機能により、全ての機器がバスの動作状況を把握することができる。

#### 【 0 0 2 6 】

又、1 3 9 4 ネットワークは、Plug&Play と呼ばれる機能を有している。この機能により、全ての機器の電源を O F F にすることなく、接続するだけで自動に接続機器を認識することができる。

#### 【 0 0 2 7 】

又、1 3 9 4 ネットワークは、1 0 0 / 2 0 0 / 4 0 0 M b p s のデータ転送速度に対応している。上位のデータ転送速度を持つ機器は、下位のデータ転送速度をサポートすることができるため、異なるデータ転送速度に対応する機器同士

を接続することができる。

#### 【0 0 2 8】

更に、1 3 9 4 ネットワークは、2 つの異なるデータ転送方式（即ち、アシンクロナス転送モードとアイソクロナス転送モード）に対応している。

#### 【0 0 2 9】

アシンクロナス（Asynchronous）転送モードは、必要に応じて非同期に転送することが要求されるデータ（即ち、コントロール信号やファイルデータ等）を転送する際に有効である。又、アイソクロナス（Isochronous）転送モードは、所定量のデータを一定のデータレートで連続的に転送することが要求されるデータ（即ち、ビデオデータやオーディオデータ等）を転送する際に有効である。

#### 【0 0 3 0】

アシンクロナス転送モードとアイソクロナス転送モードとは、各通信サイクル（通常 1 サイクルは、1 2 5  $\mu$  S）内において、混在させることが可能である。各転送モードは、サイクルの開始を示すサイクル・スタート・パケット（以下、C S P）の転送後に実行される。

#### 【0 0 3 1】

尚、各通信サイクル期間において、アイソクロナス転送モードは、アシンクロナス転送モードよりも優先順位が高く設定されている。又、アイソクロナス転送モードの転送帯域は、各通信サイクル内で保証されている。

#### 【0 0 3 2】

##### （2）アーキテクチャ

次に、図 3 を用いて 1 3 9 4 インタフェースの構成要素を説明する。

#### 【0 0 3 3】

1 3 9 4 インタフェースは、機能的に複数のレイヤ（階層）から構成されている。図 3 において、1 3 9 4 インタフェースは、I E E E 1 3 9 4 規格に準拠した通信ケーブル 3 0 1 を介して他のノードの 1 3 9 4 インタフェースと接続される。又、1 3 9 4 インタフェースは、1 つ以上の通信ポート 3 0 2 を有し、通信ポート 3 0 2 は、ハードウェア部に含まれるフィジカル・レイヤ 3 0 3 と接続される。

**【0034】**

図3において、ハードウェア部は、フィジカル・レイヤ303とリンク・レイヤ304とから構成されている。フィジカル・レイヤ303は、他のノードとの物理的、電氣的なインタフェース、バスリセットの検出とそれに伴う処理、入出力信号の符号化／復号化、バス使用权の調停等を行う。又、リンク・レイヤ304は、通信パケットの生成と送受信、サイクルタイマの制御等を行なう。

**【0035】**

又、図3において、ファームウェア部は、トランザクション・レイヤ305とシリアル・バス・マネージメント306とを含んでいる。トランザクション・レイヤ305は、アシンクロナス転送モードを管理し、各種のトランザクション（リード、ライト、ロック）を提供する。シリアル・バス・マネージメント306は、後述するCSRアーキテクチャに基づいて、自ノードの制御、自ノードの接続状態の管理、自ノードのID情報の管理、シリアルバスネットワークの資源管理を行う機能を提供する。

**【0036】**

以上、ハードウェア部とファームウェア部とが実質的に1394インタフェースを構成するものであり、それらの基本構成は、IEEE1394規格により規定されている。

**【0037】**

又、ソフトウェア部に含まれるアプリケーション・レイヤ307は、使用するアプリケーションソフトによって異なり、ネットワーク上でどのようにデータを通信するのかを制御する。例えば、デジタルVTRの動画データの場合は、AV/Cプロトコルなどの通信プロトコルによって規定されている。

**【0038】****(2-1) リンク・レイヤ304**

図4は、リンク・レイヤ304の提供可能なサービスを示す図である。図4において、リンク・レイヤ304は、次の4つのサービスを提供する。即ち、①応答ノードに対して所定のパケットの転送を要求するリンク要求(LK\_DATA.request)、②応答ノードに所定のパケットの受信を通知するリンク通知(LK\_DATA.ind

ication)、③応答ノードからのアクノリッジを受信したことを示すリンク応答(LK\_DATA.response)、④要求ノードからのアクノリッジを確認するリンク確認(LK\_DATA.confirmation)である。尚、リンク応答(LK\_DATA.response)は、ブロードキャスト通信、アイソクロナスパケットの転送の場合には存在しない。

#### 【0039】

又、リンク・レイヤ304は、上述のサービスに基づいて、上述の2種類の転送方式、即ち、アシンクロナス転送モード、アイソクロナス転送モードを実現する。

#### 【0040】

##### (2-2) トランザクション・レイヤ305

図5は、トランザクション・レイヤ305の提供可能なサービスを示す図である。図5において、トランザクション・レイヤ305は、次の4つのサービスを提供する。即ち、①応答ノードに対して所定のトランザクションを要求するトランザクション要求(TR\_DATA.request)、②応答ノードに所定のトランザクション要求の受信を通知するトランザクション通知(TR\_DATA.indication)、③応答ノードからの状態情報(ライト、ロックの場合は、データを含む)を受信したことを示すトランザクション応答(TR\_DATA.response)、④要求ノードからの状態情報を確認するトランザクション確認(TR\_DATA.confirmation)である。

#### 【0041】

又、トランザクション・レイヤ305は、上述のサービスに基づいてアシンクロナス転送を管理し、次の3種類のトランザクション、即ち、①リード・トランザクション、②ライト・トランザクション、③ロック・トランザクションを実現する。

#### 【0042】

①リード・トランザクションは、要求ノードが応答ノードの特定アドレスに格納された情報を読み取る。

#### 【0043】

②ライト・トランザクションは、要求ノードが応答ノードの特定アドレスに所定の情報を書き込む。



**【0044】**

③ロック・トランザクションは、要求ノードが応答ノードに対して参照データと更新データとを転送し、応答ノードの特定アドレスの情報とその参照データとを比較し、その比較結果に応じて特定アドレスの情報を更新データに更新する。

**【0045】****(2-3) シリアル・バス・マネージメント 306**

シリアル・バス・マネージメント 306 は、具体的に、次の 3 つの機能を提供することができる。3 つの機能とは、即ち、①ノード制御、②アイソクロナス・リソース・マネージャ（以下、IRM）、③バスマネージャである。

**【0046】**

①ノード制御は、上述の各レイヤを管理し、他のノードとの間で実行されるアシンクロナス転送を管理する機能を提供する。

**【0047】**

②IRMは、他のノードとの間で実行されるアイソクロナス転送を管理する機能を提供する。具体的には、転送帯域幅とチャンネル番号の割り当てに必要な情報を管理し、これらの情報を他のノードに対して提供する。

**【0048】**

IRMは、ローカルバス上に唯一存在し、バスリセット毎に他の候補者（IRMの機能を有するノード）の中から動的に選出される。又、IRMは、後述のバスマネージャの提供可能な機能（接続構成の管理、電源管理、速度情報の管理等）の一部を提供してもよい。

**【0049】**

③バスマネージャは、IRMの機能を有し、IRMよりも高度なバス管理機能を提供する。具体的には、より高度な電源管理（通信ケーブルを介して電源の供給が可能か否か、電源の供給が必要か否か等の情報を各ノード毎に管理）、より高度な速度情報の管理（各ノード間の最大転送速度の管理）、より高度な接続構成の管理（トポロジ・マップの作成）、これらの管理情報に基づくバスの最適化等を行ない、更にこれらの情報を他のノードに提供する機能を有する。

**【0050】**

又、バスマネージャは、シリアルバスネットワークを制御するためのサービスをアプリケーションに対して提供できる。ここで、サービスには、シリアルバス制御要求 (SB\_CONTROL.request)、シリアルバス・イベント制御確認 (SB\_CONTROL.confirmation)、シリアルバス・イベント通知 (SB\_CONTROL.indication) 等がある。

#### 【 0 0 5 1 】

SB\_CONTROL.requestは、アプリケーションがバスリセットを要求するサービスである。SB\_CONTROL.confirmationは、SB\_CONTROL.requestをアプリケーションに対して確認するサービスである。SB\_CONTROL.indicationは、非同期に発生するイベントをアプリケーションに対して通知するサービスである。

#### 【 0 0 5 2 】

##### (3) アドレス指定

図 6 は、1 3 9 4 インタフェースにおけるアドレス空間を説明する図である。尚、1 3 9 4 インタフェースは、ISO/IEC 13213:1994に準じたC S R (Command and Status Register) アーキテクチャに従い、6 4 ビット幅のアドレス空間を規定している。

#### 【 0 0 5 3 】

図 6 において、最初の1 0 ビットのフィールド6 0 1 は、所定のバスを指定するID番号に使用され、次の6 ビットのフィールド6 0 2 は、所定の機器 (ノード) を指定するID番号に使用される。この上位1 6 ビットを「ノードID」と呼び、各ノードはこのノードIDにより他のノードを識別する。又、各ノードは、このノードIDを用いて相手を識別した通信を行うことができる。

#### 【 0 0 5 4 】

残りの4 8 ビットからなるフィールドは、各ノードの具備するアドレス空間 (2 5 6 Mバイト構造) を指定する。その内の2 0 ビットのフィールド6 0 3 は、アドレス空間を構成する複数の領域を指定する。

#### 【 0 0 5 5 】

フィールド6 0 3 において、「0 ~ 0 x F F F F D」の部分は、メモリ空間と呼ばれる。「0 x F F F F E」の部分は、プライベート空間と呼ばれ、各ノード

で自由に利用できるアドレスである。又、「0 x F F F F E」の部分は、レジスタ空間と呼ばれ、バスに接続されたノード間において共通の情報を格納する。各ノードは、レジスタ空間の情報をを用いることにより、各ノード間の通信を管理することができる。

#### 【0056】

最後の28ビットのフィールド604は、各ノードにおいて共通或いは固有となる情報が格納されるアドレスを指定する。

#### 【0057】

例えば、レジスタ空間において、最初の512バイトは、CSRアーキテクチャのコア（CSRコア）レジスタ用に使用される。CSRコア・レジスタに格納される情報のアドレス及び機能を図7に示す。図中のオフセットは、「0 x F F F F 0 0 0 0 0 0 0」からの相対位置である。

#### 【0058】

次の512バイトは、シリアルバス用のレジスタとして使用される。シリアルバス・レジスタに格納される情報のアドレス及び機能を図8に示す。図中のオフセットは、「0 x F F F F F 0 0 0 0 2 0 0」からの相対位置である。

#### 【0059】

その次の1024バイトは、コンフィグレーションROM（Configuration ROM）用に使用される。

#### 【0060】

コンフィグレーションROMには最小形式と一般形式とがあり、「0 x F F F F F 0 0 0 0 4 0 0」から配置される。最小形式のコンフィグレーションROMを図9に示す。図9において、ベンダIDは、IEEEにより各ベンダに対して固有に割り当てられた24ビットの数値である。

#### 【0061】

又、一般形式のコンフィグレーションROMを図10に示す。図10において、上述のベンダIDは、ルートディレクトリ（Root Directory）1002に格納されている。Bus Info Block1001とRoot Leaf1005とには、各ノードを識別する固有のID情報としてノードユニークIDを保持することが可能である

。

#### 【0062】

ここで、ノードユニークIDは、メーカー、機種に関わらず、1つのノードを特定することのできる固有のIDを定めるようになっている。ノードユニークIDは64ビットにより構成され、上位24ビットは上述のベンダIDを示し、下位48ビットは各ノードを製造するメーカーにおいて自由に設定可能な情報（例えば、ノードの製造番号等）を示す。尚、このノードユニークIDは、例えばバスリセットの前後で継続して特定のノードを認識する場合に使用される。

#### 【0063】

又、図10において、ルートディレクトリ1002には、ノードの基本的な機能に関する情報を保持することが可能である。詳細な機能情報は、ルートディレクトリ1002からオフセットされるサブディレクトリとしてのユニットディレクトリ（Unit Directories）1004に格納される。ユニットディレクトリ1004には、例えば、ノードのサポートするソフトウェアユニットに関する情報が格納される。具体的には、ノード間のデータ通信を行うためのデータ転送プロトコル、所定の通信手順を定義するコマンドセット等に関する情報が保持される。

#### 【0064】

又、図10において、ノード依存情報ディレクトリ（Node Dependent Info Directory）1003には、デバイス固有の情報を保持することが可能である。ノード依存情報ディレクトリ1003は、ルートディレクトリ1002によりオフセットされる。

#### 【0065】

更に、図10において、ベンダー依存情報（Vendor Dependent Information）1006には、ノードを製造、或いは販売するベンダ固有の情報を保持することができる。

#### 【0066】

残りの領域は、ユニット空間と呼ばれ、各ノード固有の情報、例えば、各機器の識別情報（会社名、機種名等）や使用条件等が格納されたアドレスを指定する。ユニット空間のシリアルバス装置レジスタに格納される情報のアドレス及び機

能を図 11 に示す。図中のオフセットは、「0 x F F F F F 0 0 0 0 8 0 0」からの相対位置である。

#### 【0067】

尚、一般的に、異種のバスシステムの設計を簡略化したい場合、各ノードは、レジスタ空間の最初の 2048 バイトのみを使うべきである。つまり、CSR コア・レジスタ、シリアルバス・レジスタ、コンフィグレーション ROM、ユニット空間の最初の 2048 バイトの合わせて 4096 バイトで構成することが望ましい。

#### 【0068】

##### (4) 通信ケーブルの構成

図 12 に IEEE 1394 規格に準拠した通信ケーブルの断面図を示す。

#### 【0069】

通信ケーブルは、2 組のツイストペア信号線と電源ラインとにより構成されている。電源ラインを設けることによって、1394 インタフェースは、主電源の OFF となった機器、故障により電力低下した機器等にも電力を供給することができる。尚、電源線内を流れる電源の電圧は 8 ~ 40 V、電流は最大電流 DC 1.5 A と規定されている。

#### 【0070】

2 組のツイストペア信号線には、DS-Link (Data/Strobe Link) 符号化方式にて符号化された情報信号が伝送される。図 13 は、DS-Link 符号化方式を説明する図である。

#### 【0071】

この DS-Link 符号化方式は、高速なシリアルデータ通信に適しており、その構成は、2 組のより対線を必要とする。一組のより対線は、データ信号を送り、他のより対線は、ストロブ信号を送る構成になっている。受信側は、2 組の信号線から受信したデータ信号とストロブ信号との排他的論理和をとることによって、クロックを再現することができる。

#### 【0072】

尚、DS-Link 符号化方式を用いることにより、1394 インタフェースには、

例えば次のような利点がある。①他の符号化方式に比べて転送効率が高い。②PLL回路が不要となり、コントローラLSIの回路規模を小さくできる。③アイドル状態であることを示す情報を送る必要が無いため、トランシーバ回路をスリープ状態とし易く、消費電力の低減を図ることができる。

#### 【0073】

##### (5) バスリセット

各ノードの1394インタフェースは、ネットワークの接続構成に変化が生じたことを自動的に検出することができる。この場合、1394ネットワークは以下に示す手順によりバスリセットと呼ばれる処理を行う。尚、接続構成の変化は、各ノードの具備する通信ポートにかかるバイアス電圧の変化により検知することができる。

#### 【0074】

ネットワークの接続構成の変化（例えば、ノードの挿抜、ノードの電源のON/OFFなどによるノード数の増減）を検出したノード、又は新たな接続構成を認識する必要があるノードは、1394インタフェースを介して、バス上にバスリセット信号を送信する。

#### 【0075】

バスリセット信号を受信したノードの1394インタフェースは、バスリセットの発生を自身のリンク・レイヤ304に伝達すると共に、そのバスリセット信号を他のノードに転送する。バスリセット信号を受信したノードは、今まで認識していたネットワークの接続構成及び各機器に割り当てられたノードIDをクリアにする。最終的に全てのノードがバスリセット信号を検知した後、各ノードは、バスリセットに伴う初期化処理（即ち、新たな接続構成の認識と新たなノードIDの割り当て）を自動的に行う。

#### 【0076】

尚、バスリセットは、先に述べたような接続構成の変化による起動の他に、ホスト側の制御によって、アプリケーション・レイヤ307がフィジカル・レイヤ303に対して直接命令を出すことによって起動させることも可能である。

#### 【0077】

又、バスリセットが起動するとデータ転送は一時中断され、バスリセットに伴う初期化処理の終了後、新しいネットワークのもとで再開される。

#### 【0078】

##### (6) バスリセット起動後のシーケンス

バスリセットの起動後、各ノードの1394インタフェースは、新たな接続構成の認識と新たなノードIDの割り当てとを自動的に実行する。以下、バスリセットの開始からノードIDの割り当て処理までの基本的なシーケンスを図14～16を用いて説明する。

#### 【0079】

図14は、図2の1394ネットワークにおけるバスリセット起動後の状態を説明する図である。

#### 【0080】

図14において、ノードAは1つの通信ポート、ノードBは2つの通信ポート、ノードCは2つの通信ポート、ノードDは3つの通信ポート、ノードEは1つの通信ポート、ノードFは1つの通信ポートを具備している。各ノードの通信ポートには、各ポートを識別するためにポート番号が付されている。

#### 【0081】

以下、図14におけるバスリセットの開始からノードIDの割り当てまでを図15のフローチャートを用いて説明する。

#### 【0082】

図15において、1394ネットワークを構成する各ノードA～Fは、バスリセットが発生したか否かを常時監視している（ステップS1501）。接続構成の変化を検出したノードからバスリセット信号が出力されると、各ノードは以下の処理を実行する。

#### 【0083】

バスリセットの発生後、各ノードは、夫々の具備する通信ポート間において親子関係の宣言を行なう（ステップS1502）。

#### 【0084】

各ノードは、全てのノード間の親子関係が決定されるまで、ステップS150

2 の処理を繰り返し行なう（ステップ S 1 5 0 3）。

【0085】

全てのノード間の親子関係が決定した後、1394 ネットワークは、ネットワークの調停を行なうノード、即ちルートを決定する。（ステップ S 1 5 0 4）。

【0086】

ルートを決定した後、各ノードの1394 インタフェース々は、自己のノード ID を自動的に設定する作業を実行する（ステップ S 1 5 0 5）。

【0087】

全てのノードに対してノード ID の設定がなされるまで、各ノードは所定の手順に基づきステップ S 1 5 0 5 の処理を実行する（ステップ S 1 5 0 6）。

【0088】

最終的に全てのノードに対してノード ID が設定された後、各ノードは、アイソクロナス転送或いはアシンクロナス転送を実行する（ステップ S 1 5 0 7）。

【0089】

ステップ S 1 5 0 7 の処理を実行すると共に、各ノードの1394 インタフェースは、再びバスリセットの発生を監視する。バスリセットが発生した場合には、ステップ S 1 5 0 1 以降の処理を再び実行する。

【0090】

以上の手順により、各ノードの1394 インタフェースは、バスリセットが起動する毎に、新たな接続構成の認識と新たなノード ID の割り当てとを自動的に実行することができる。

【0091】

（7）親子関係の決定

次に、図 1 6 を用いて、図 1 5 に示したステップ S 1 5 0 2 の処理（即ち、各ノード間の親子関係を認識する処理）について詳細に説明する。

【0092】

図 1 6 において、バスリセットの発生後、1394 ネットワーク上の各ノード A ～ F は、自分の具備する通信ポートの接続状態（接続又は未接続）を確認する（ステップ S 1 6 0 1）。



**【0093】**

通信ポートの接続状態の確認後、各ノードは、他のノードと接続されている通信ポート（以下、接続ポート）の数をカウントする（ステップS1602）。

**【0094】**

ステップS1602の処理の結果、接続ポートの数が1つである場合、そのノードは、自分が「リーフ」とであると認識する（ステップS1603）。ここで、リーフとは、1つのノードとのみ接続されているノードのことである。

**【0095】**

リーフとなるノードは、その接続ポートに接続されているノードに対して、「自分は子（Child）」であることを宣言する（ステップS1604）。このとき、リーフは、その接続ポートが「親ポート（親ノードと接続された通信ポート）」であると認識する。

**【0096】**

ここで、親子関係の宣言は、まず、ネットワークの末端であるリーフとブランチとの間にて行われ、続いて、ブランチとブランチとの間で順次に行われる。各ノード間の親子関係は、早く宣言の行なえる通信ポートから順に決定される。又、各ノード間において、子であることを宣言した通信ポートは「親ポート」とであると認識され、その宣言を受けた通信ポートは「子ポート（子ノードと接続された通信ポート）」であると認識される。例えば、図14において、ノードA、E、Fは、自分がリーフであると認識した後、親子関係の宣言を行う。これにより、ノードA-B間では子-親、ノードE-D間では子-親、ノードF-D間では子-親と決定される。

**【0097】**

又、ステップS1602の処理の結果、接続ポートの数が2つ以上の場合、そのノードは、自分を「ブランチ」とであると認識する（ステップS1605）。ここで、ブランチとは、2つ以上のノードと接続されているノードのことである。

**【0098】**

ブランチとなるノードは、各接続ポートのノードから親子関係の宣言を受け付ける（ステップS1606）。宣言を受け付けた接続ポートは、「子ポート」と

して認識される。

#### 【0099】

1つの接続ポートを「子ポート」と認識した後、ブランチは、まだ親子関係の決定されていない接続ポート（即ち、未定義ポート）が2つ以上あるか否かを検出する（ステップS1607）。その結果、未定義ポートが2つ以上ある場合、ブランチは、再びステップS1606の動作を行う。

#### 【0100】

ステップS1607の結果、未定義ポートが1つだけ存在する場合、ブランチは、その未定義ポートが「親ポート」とであると認識し、そのポートに接続されているノードに対して「自分は子」であることを宣言する（ステップS1608、S1609）。

#### 【0101】

ここで、ブランチは、残りの未定義ポートが1つになるまで自分自身が子であると他のノードに対して宣言することができない。例えば、図14において、ノードB、C、Dは、自分がブランチであると認識すると共に、リーフ或いは他のブランチからの宣言を受け付ける。ノードDは、D-E間、D-F間の親子関係が決定した後、ノードCに対して親子関係の宣言を行っている。又、ノードDからの宣言を受けたノードCは、ノードBに対して親子関係の宣言を行っている。

#### 【0102】

又、ステップS1608の処理の結果、未定義ポートが存在しない場合（つまり、ブランチの具備する全ての接続ポートが親ポートとなった場合）、そのブランチは、自分自身がルートであることを認識する。（ステップS1610）。

#### 【0103】

例えば、図14において、接続ポートの全てが親ポートとなったノードBは、1394ネットワーク上の通信を調停するルートとして他のノードに認識される。ここで、ノードBがルートと決定されたが、ノードBの親子関係を宣言するタイミングが、ノードCの宣言するタイミングに比べて早い場合には、他のノードがルートになる可能性もある。即ち、宣言するタイミングによっては、どのノードもルートとなる可能性がある。従って、同じネットワーク構成であっても同じ

ノードがルートになるとは限らない。

#### 【0104】

このように全ての接続ポートの親子関係が宣言されることによって、各ノードは、1394ネットワークの接続構成を階層構造（ツリー構造）として認識することができる（ステップS1611）。尚、上述の親ノードは階層構造における上位であり、子ノードは階層構造における下位となる。

#### 【0105】

##### （8）ノードIDの割り当て

図17は、図15に示したステップS1505の処理（即ち、自動的に各ノードのノードIDを割り当てる処理）を詳細に説明するフローチャートである。ここで、ノードIDは、バス番号とノード番号とから構成されるが、本実施の形態では、各ノードを同一バス上に接続するものとし、各ノードには同一のバス番号が割り当てられるものとする。

#### 【0106】

図17において、ルートは、ノードIDが未設定のノードが接続されている子ポートの内、最小番号を有する通信ポートに対してノードIDの設定許可を与える（ステップS1701）。

#### 【0107】

尚、図17において、ルートは、最小番号の子ポートに接続されている全ノードのノードIDを設定した後、その子ポートを設定済とし、次に最小となる子ポートに対して同様の制御を行なう。最終的に子ポートに接続された全てのノードのID設定が終了した後、ルート自身のノードIDを設定する。尚、ノードIDに含まれるノード番号は、基本的にリーフ、ブランチの順に0、1、2…と割り当てられる。従って、ルートが最も大きなノード番号を有することになる。

#### 【0108】

ステップS1701において、設定許可を得たノードは、自分の子ポートの内、ノードIDが未設定となるノードを含む子ポートがあるか否かを判断する（ステップS1702）。

#### 【0109】

ステップS1702において、未設定ノードを含む子ポートが検出された場合、上述の設定許可を得たノードは、その子ポートに直接接続されたノードに対してその設定許可を与えるように制御する（ステップS1703）。

#### 【0110】

ステップS1703の処理後、上述の設定許可を得たノードは、自分の子ポートの内、ノードIDが未設定であるノードを含む子ポートがあるか否かを判断する（ステップS1704）。ここで、ステップS1704の処理後、未設定ノードを含む子ポートの存在が検出された場合、そのノードは、再びステップS1703の処理を実行する。

#### 【0111】

又、ステップS1702或いはS1704において、未設定ノードを含む子ポートが検出されなかった場合、設定許可を得たノードは、自分自身のノードIDを設定する（ステップS1705）。

#### 【0112】

自分のノードIDを設定したノードは、自己のノード番号、通信ポートの接続状態に関する情報等を含んだセルフIDパケットをブロードキャストする（ステップS1706）。尚、ブロードキャストとは、あるノードの通信パケットを、1394ネットワークを構成する不特定多数のノードに対して転送することである。

#### 【0113】

ここで、各ノードは、このセルフIDパケットを受信することにより、各ノードに割り当てられたノート番号を認識することができ、自分に割り当てられるノード番号を知ることができる。例えば、図14において、ルートであるノードBは、最小ポート番号「#1」の通信ポートに接続されたノードAに対してノードID設定の許可を与える。ノードAは、自己のノード番号「No. 0」と割り当て、自分自身に対してバス番号とノード番号とからなるノードIDを設定する。又、ノードAは、そのノード番号を含むセルフIDパケットをブロードキャストする。

#### 【0114】

図18にセルフIDパケットの構成例を示す。図18において、1801はセルフIDパケットを送出したノードのノード番号を格納するフィールド、1802は対応可能な転送速度に関する情報を格納するフィールド、1803はバス管理機能（バスマネージャの能力の有無等）の有無を示すフィールド、1804は電力の消費及び供給の特性に関する情報を格納するフィールドである。

#### 【0115】

又、図18において、1805はポート番号「#0」となる通信ポートの接続状態に関する情報（接続、未接続、通信ポートの親子関係等）を格納するフィールド、1806はポート番号「#1」となる通信ポートの接続状態に関する情報（接続、未接続、通信ポートの親子関係等）を格納するフィールド、1807はポート番号「#2」となる通信ポートの接続状態に関する情報（接続、未接続、通信ポートの親子関係等）を格納するフィールドである。

#### 【0116】

尚、セルフIDパケットを送出するノードにバスマネージャとなり得る能力がある場合には、フィールド1803に示すコンテナビットを「1」とし、なり得る能力がなければ、コンテナビットを0とする。

#### 【0117】

ここで、バスマネージャとは、上述のセルフIDパケットに含まれる各種の情報に基づいて、バスの電源管理（通信ケーブルを介して電源の供給が可能か否か、電源の供給が必要か否か等の情報を各ノード毎に管理する）、速度情報の管理（各ノードの対応可能な転送速度に関する情報から各ノード間の最大転送速度を管理する）、トポロジ・マップ情報の管理（通信ポートの親子関係情報からネットワークの接続構成を管理する）、トポロジ・マップ情報に基づくバスの最適化等を行ない、それらの情報を他のノードに提供する機能を有するノードである。これらの機能により、バスマネージャとなるノードは1394ネットワーク全体のバス管理を行なうことができる。

#### 【0118】

ステップS1706の処理後、ノードIDの設定を行ったノードは、親ノードがあるか否かを判断する（ステップS1707）。親ノードがある場合、その親

ノードが、ステップ S 1702 以下の処理を再び実行する。そして、まだノード ID の設定されていないノードに対して許可を与える。

#### 【0119】

又、親ノードが存在しない場合、そのノードは、ルート自身であると判断される。ルートは、全ての子ポートに接続されたノードに対してノード ID が設定されたか否かを判別する（ステップ S 1708）。

#### 【0120】

ステップ S 1708 において、全てのノードに対する ID 設定処理が終了しなかった場合、ルートは、そのノードを含む子ポートの内、最小番号となる子ポートに対して ID 設定の許可を与える（ステップ S 1701）。その後、ステップ S 1702 以下の処理を実行する。

#### 【0121】

又、全てのノードに対する ID 設定処理が終了した場合、ルートは、自分自身のノード ID の設定を実行する（ステップ S 1709）。ノード ID の設定後、ルートは、セルフ ID パケットをブロードキャストする（ステップ S 1710）。

#### 【0122】

以上の処理によって、1394 ネットワークは、各ノードに対して自動的にノード ID を割り当てることができる。

#### 【0123】

ここで、ノード ID の設定処理後、複数のノードがバスマネージャの能力を具備する場合、ノード番号の最も大きいノードがバスマネージャとなる。つまり、ネットワーク内で最大となるノード番号を持つルートがバスマネージャになり得る機能を有している場合には、ルートがバスマネージャとなる。

#### 【0124】

しかしながら、ルートにその機能が備わっていない場合には、ルートの次に大きいノード番号を具備するノードがバスマネージャとなる。又、どのノードがバスマネージャになったかについては、各ノードがブロードキャストするセルフ ID パケット内のコンテンダビット 1803 をチェックすることにより把握するこ

とができる。

#### 【0125】

##### (9) アービトレーション

図19は、図1の1394ネットワークにおけるアービトレーションを説明する図である。

#### 【0126】

1394ネットワークでは、データ転送に先立って、必ずバス使用权のアービトレーション（調停）を行なう。1394ネットワークは、論理的なバス型ネットワークであり、各ノードから転送された通信パケットを他のノードに中継することによって、ネットワーク内の全てのノードに同じ通信パケットを転送することのできる。従って、通信パケットの衝突を防ぐために、必ずアービトレーションが必要となる。これによって、ある時間において一つのノードのみが転送を行なうことができる。

#### 【0127】

図19(a)は、ノードBとノードFとが、バス使用权の要求を発している場合について説明する図である。

#### 【0128】

アービトレーションが始まるとノードB、Fは、夫々親ノードに向かって、バス使用权の要求を発する。ノードBの要求を受けた親ノード（即ち、ノードC）は、自分の親ノード（即ち、ノードD）に向かって、そのバス使用权を中継する。この要求は、最終的に調停を行なうルート（ノードD）に届けられる。

#### 【0129】

バス使用要求を受けたルートは、どのノードにバスを使用させるかを定める。この調停作業はルートとなるノードのみが行なえるものであり、調停によって勝ったノードにはバスの使用許可が与えられる。

#### 【0130】

図19(b)は、ノードFの要求が許可され、ノードBの要求が拒否されたことを示す図である。

#### 【0131】

アービトレーションに負けたノードに対してルートは、DP (Data prefix) パケットを送り、拒否されたことを知らせる。拒否されたノードは、次のアービトレーションまでバス使用要求を待機する。

#### 【0132】

以上のようにアービトレーションを制御することによって、1394 ネットワークは、バスの使用权を管理することができる。

#### 【0133】

##### (10) 通信サイクル

アイソクロナス転送モードとアシンクロナス転送モードとは、各通信サイクル期間内において時分割に混在させることができる。ここで、通信サイクルの期間は、通常、 $125\mu\text{S}$ である。

#### 【0134】

図20は、1通信サイクルにおいてアイソクロナス転送モードとアシンクロナス転送モードとを混在させた場合を説明する図である。

#### 【0135】

アイソクロナス転送モードは、アシンクロナス転送モードより優先して実行される。その理由は、サイクル・スタート・パケットの後、アシンクロナス転送を起動するために必要なアイドル期間 (subaction gap) が、アイソクロナス転送を起動するため必要なアイドル期間 (isochronous gap) よりも長くなるように設定されているためである。これにより、アイソクロナス転送は、アシンクロナス転送に優先して実行される。

#### 【0136】

図20において、各通信サイクルのスタート時には、サイクル・スタート・パケット (以下、CSP) が所定のノードから転送される。各ノードは、このCSPを用いて時刻調整を行うことによって、他のノードと同じ時間を計時することができる。

#### 【0137】

##### (11) アイソクロナス転送モード

アイソクロナス転送モードは、同期型の転送方式である。アイソクロナスモー



ド転送は、通信サイクルの開始後、所定の期間において実行可能である。又、アイソクロナス転送モードは、リアルタイム転送を維持するために、各サイクル毎に必ず実行される。

#### 【0 1 3 8】

アイソクロナス転送モードは、特に動画像データや音声データ等のリアルタイムな転送を必要とするデータの転送に適した転送モードである。アイソクロナス転送モードは、アシンクロナス転送モードのように1対1の通信ではなく、ブロードキャスト通信である。つまり、あるノードから送出されたパケットは、ネットワーク上の全てのノードに対して一様に転送される。尚、アイソクロナス転送には、ack（受信確認用返信コード）は存在しない。

#### 【0 1 3 9】

図20において、チャンネルe（ch e）、チャンネルs（ch s）、チャンネルk（ch k）は、各ノードがアイソクロナス転送を行う期間を示す。1394インタフェースでは、複数の異なるアイソクロナス転送を区別するために、夫々異なるチャンネル番号を与えている。これにより、複数ノード間でのアイソクロナス転送が可能となる。ここで、このチャンネル番号は、送信先を特定するものではなく、データに対する論理的な番号を与えているに過ぎない。

#### 【0 1 4 0】

又、図20に示したアイソクロナス gapとは、バスのアイドル状態を示すものである。このアイドル状態が一定時間を経過した後、アイソクロナス転送を希望するノードは、バスが使用できると判断し、アービトレーションを実行する。

#### 【0 1 4 1】

次に、図21にアイソクロナス転送モードに基づいて転送される通信パケットのフォーマットを示す。以下、アイソクロナス転送モードに基づいて転送される通信パケットを、アイソクロナスパケットと称する。

#### 【0 1 4 2】

図21において、アイソクロナスパケットはヘッダ部2101、ヘッダCRC 2102、データ部2103、データCRC 2104から構成される。

#### 【0 1 4 3】

ヘッダ部 2 1 0 1 には、データ部 2 1 0 3 のデータ長を格納するフィールド 2 1 0 5、アイソクロナスパケットのフォーマット情報を格納するフィールド 2 1 0 6、アイソクロナスパケットのチャネル番号を格納するフィールド 2 1 0 7、パケットのフォーマット及び実行しなければならない処理を識別するトランザクションコード (tcode) を格納するフィールド 2 1 0 8、同期化コードを格納するフィールド 2 1 0 9 がある。

#### 【0 1 4 4】

##### (1 2) アシンクロナス転送モード

アシンクロナス転送モードは、非同期型の転送方式である。アシンクロナス転送は、アイソクロナス転送期間の終了後、次の通信サイクルが開始されるまでの間（即ち、次の通信サイクルの C S P が転送されるまでの間）、実行可能である。

#### 【0 1 4 5】

図 2 0 において、最初のサブアクション・ギャップ (subaction gap) は、バスのアイドル状態を示すものである。このアイドル時間が一定値になった後、アシンクロナス転送を希望するノードは、バスが使用できると判断し、アービトレーションを実行する。

#### 【0 1 4 6】

アービトレーションによりバスの使用权を得たノードは、図 2 2 に示すパケットを所定のノードに対して転送する。このパケットを受信したノードは、ack（受信確認用返送コード）或いは応答パケットを ack gap 後に返送する。

#### 【0 1 4 7】

図 2 2 は、アシンクロナス転送モードに基づく通信パケットのフォーマットを示す図である。以下、アシンクロナス転送モードに基づいて転送される通信パケットを、アシンクロナスパケットと称する。

#### 【0 1 4 8】

図 2 2 において、アシンクロナスパケットは、ヘッダ部 2 2 0 1、ヘッダ CRC 2 2 0 2、データ部 2 2 0 3、データ CRC 2 2 0 4 から構成される。

#### 【0 1 4 9】

ヘッダ部 2 2 0 1 において、フィールド 2 2 0 5 には宛先となるノードのノード ID、フィールド 2 2 0 6 にはソースとなるノードのノード ID、フィールド 2 2 0 7 には一連のトランザクションを示すためのラベル、フィールド 2 2 0 8 には再送ステータスを示すコード、フィールド 2 2 0 9 にはパケットのフォーマット及び実行しなければならない処理を識別するトランザクションコード (tcod e)、フィールド 2 2 1 0 には優先順位、フィールド 2 2 1 1 には宛先のメモリ・アドレス、フィールド 2 2 1 2 にはデータ部のデータ長、フィールド 2 2 1 3 には拡張されたトランザクション・コードが格納される。

#### 【 0 1 5 0 】

又、アシンクロナス転送は、自己ノードから相手ノードへの 1 対 1 の通信である。転送元ノードから転送されたパケットは、ネットワーク中の各ノードに行き渡るが、自分宛てのアドレス以外のものは無視される。従って、宛先となるノードのみが、そのパケットを読み込むことができる。

#### 【 0 1 5 1 】

尚、アシンクロナス転送中に次の C S P を転送すべき時間に至った場合、無理に転送を中断せず、その転送が終了した後、次の C S P を送信する。これにより、1 つの通信サイクルが 1 2 5  $\mu$  S 以上続いたときは、その分、次の通信サイクル期間を短縮する。このようにすることによって、1 3 9 4 ネットワークは、ほぼ一定の通信サイクルを保持することができる。

#### 【 0 1 5 2 】

##### ( 1 3 ) デバイスマップ

デバイスマップを作成するためにアプリケーションが 1 3 9 4 ネットワークのトポロジーを知る手段として、I E E E 1 3 9 4 規格上は以下の手段がある。

#### 【 0 1 5 3 】

1. バスマネージャのトポロジーマップレジスターをリードする
2. バスリセット時にセルフ ID パケットから推定する

しかし上記 1、2 の手段では、各ノードの親子関係によるケーブル接続順のトポロジーは判明するものの、物理的な位置関係のトポロジーを知ることはできない(実装されていないポートまで見えてしまう、といった問題もある)。

**【0154】**

また、デバイスマップを作成するための情報を、コンフィギュレーションROM以外のデータベースとして持つ、といった手段もあるが、その場合、各種情報を得る手段はデータベースアクセスのためのプロトコルに依存してしまう。

**【0155】**

ところで、コンフィギュレーションROM自体やコンフィギュレーションROMを読む機能は、IEEE1394規格を遵守したデバイスが必ず持つものである。そこで、デバイスの位置、機能等の情報を各ノードのコンフィギュレーションROMに格納し、それらをアプリケーションから読む機能を与えることにより、データベースアクセス、データ転送等の特定のプロトコルに依存することなく、各ノードのアプリケーションがいわゆるデバイスマップ表示機能を実装することができる。

**【0156】**

すなわち、コンフィギュレーションROMにはノード固有の情報として物理的な位置、機能などが格納可能であり、デバイスマップ表示機能の実現に使用することが可能である。

**【0157】**

この場合、アプリケーションが物理的な位置関係による1394ネットワークトポロジーを知る手段としては、バスリセット時やユーザからの要求時に、各ノードのコンフィギュレーションROMを読み取ることにより、1394ネットワークのトポロジーを知る、という方法が可能となる。さらに、コンフィギュレーションROM内にノードの物理的位置のみならず、機能などの各種ノード情報も記述すれば、コンフィギュレーションROMを読むことで、ノードの物理的位置と同時に各ノードの機能情報等も得ることができる。アプリケーションが各ノードのコンフィギュレーションROM情報を取得する際には、指定ノードの任意のコンフィギュレーションROM情報を取得するAPIを用いる。

**【0158】**

このような手段を用いることにより、IEEE1394ネットワーク上のデバイスのアプリケーションは、物理的なトポロジーマップ、各ノードの機能マップ

など、用途に応じて様々なデバイスマップを作成することができ、ユーザが必要な機能をもつデバイスを選択する、といったことも可能となる。

#### 【0159】

##### (14) SBP-2 の概要

SBP-2 (Serial Bus Protocol 2) は、NCITS傘下のTechnical Committee T10のプロジェクトとして1996より標準化の審議が進められ、1998年にANSI NCITS 325-1998として標準化が認可された、IEEE 1394バスに関するプロトコルである。レイヤとしては、IEEE 1394-1995のトランザクション層の上位に位置するコマンド／データ転送プロトコルである。当初SBP-2は、SCSIコマンドによって、IEEE 1394上のデバイスを動作させることを、主な目的として開発されたが、SCSIコマンドに限らず、他のコマンドを載せることもできる。

#### 【0160】

SBP-2の特徴として以下の4つが挙げられる。

#### 【0161】

1) イニシエータ (Initiator) ／ターゲット (Target) 構成のマスタスレーブモデル (Master Slave Model) であり、マスタであるイニシエータがログイン、ログアウト、タスク、マネージメント、コマンド発行等の全ての権限と責任を持つ。

#### 【0162】

2) バスモデルとしてのIEEE 1394の特徴を生かした共有メモリモデル (Shared Memory Model) である。コマンド等のターゲットへの要求内容は、基本的には全てシステムメモリ内に置かれ、要求を受けたターゲットがシステムメモリの内容を読みに行く。又は、システムメモリへ要求されたステータス等の情報を書きこむ。

#### 【0163】

即ち、通信のためのリソースを一箇所に集中できるので、リソースの負担を非常に軽くでき、かつターゲットが自分のペースでシステムメモリへ読み書きできるので、ターゲットの設計の自由度が高く、システムメモリのアクセスをH/W

化することにより高速化が容易である。つまり、高性能にも、徹底した低コストモデルにもできる。

#### 【0164】

3) メッセージ交換のためのコマンド群を一連のリンクリストとして記述する仕組みがあるため、レイテンシによる効率低下を隠蔽できるため、IEEE1394バスの特徴を生かした、高速で効率の高いデータ通信を実現できる。

#### 【0165】

4) コマンドセットに依存しない構造である。つまり様々なコマンドセットに対応可能である。

#### 【0166】

特徴1) に示したように、SBP-2はマスタであるイニシエータが全ての権限と責任を持つマスタスレイブモデルであるため、全てはイニシエータからの動作をトリガーにして行われる。イニシエータからのログイン、ログアウト要求やタスクマネージメント要求、コマンド等は、ORB(Operation Request Block)と呼ぶデータ構造に内包された形でターゲットに送られる。正しくは、イニシエータが自メモリに置き、ターゲットがそれを読み出す。図23に、主なORBの種類を示す。

#### 【0167】

1) ダミーORB：ターゲットフェッチエージェントのイニシャライズ時、タスクのキャンセル等に使用する。ターゲットにはno operationとして扱われる。

#### 【0168】

2) コマンドブロックORB：データ転送コマンド、デバイス制御コマンド等のコマンドを内包するORBである。詳細を図24に示す。対応するデータバッファ又はページテーブルのアドレス及びサイズを示すデータディスクリプタ(data\_descriptor)、データサイズフィールド(data\_size)を有する。また、次のコマンドブロックORBのアドレスを示すネクストORBフィールド(next\_ORB)を有するので、コマンドをリンクできるのが特徴である。

#### 【0169】

3) マネージメントORB：マネージメント要求(ログイン、ログアウトを含

むアクセス要求、及びタスクマネジメント要求)を内包するORBである。詳細を図25に示す。タスクマネジメントの要求内容(アボートタスクセット(Abort Task Set)、ターゲットリセット等)を示すファンクションフィールド(function)と、ターゲットからの完了ステータスのアドレスを示すステータスFIFO(Status\_FIF0)のフィールドを有するのが特徴である。

#### 【0170】

この内、マネジメントORBについては、イニシエータは、ターゲットが応答を返すまで次のORBを発行することができない。一方、ダミーORBを含むコマンドブロックORBには、図24のようにネクストORBフィールドとして次のORBのアドレスを指示するフィールドがあるため、次々とコマンドを連結し、図26のようにリンクリストの形で一連のコマンド列を発行することができる。このネクストORBフィールドがnullの場合は、次に続くORBがないことを示す。

#### 【0171】

また、このコマンドブロックORBの他のフィールドにはデータバッファ又はページテーブルのアドレス及びサイズを示すフィールド(データディスクリプタ及びデータサイズ)があり、例えばコマンドの内容がライトコマンドならば、ターゲットはデータディスクリプタで示されたシステムメモリ上のデータバッファにアクセスし、そこからライトデータを読みこむ。また、コマンドの内容がリードコマンドならば、ターゲットはデータディスクリプタで示されたシステムメモリ上のデータバッファにアクセスし、そこへコマンドの要求するデータを書きこむ。

#### 【0172】

図27及び図28に、データバッファを直接示す場合と、ページテーブルを経由する場合を示す。物理的に不連続な領域のデータバッファを、ページテーブルにより、連続的に扱うことができ、仮想記憶による連続論理領域を物理的に再配置する必要がなくなるわけである。

#### 【0173】

イニシエータからの様々な要求を実行するターゲット側の仕組みを、エージェ

ント (Agent) と称する。エージェントには、マネージメントORBを実行するマネージメントエージェントと、コマンドブロックORBを実行するコマンドブロックエージェントがある。

#### 【0174】

マネージメントエージェントには、マネージメントORBのアドレスをイニシエータがターゲットに知らせるためにストアする、マネージメントエージェントレジスタがある。

#### 【0175】

コマンドブロックエージェントは、イニシエータのコマンドリンクリストからコマンドをフェッチしてくるため、フェッチエージェントとも呼ばれる。フェッチエージェントにも、コマンドブロックORBの先頭アドレスをイニシエータがストアするORBポインタレジスタと、イニシエータがターゲットにコマンドをFetchして貰いたいことを知らせるドアベルレジスタ等を含む、コマンドブロックエージェントレジスタがある。

#### 【0176】

ターゲットはイニシエータからのORBの実行を完了すると、その実行完了のステータスを、ステータスブロックと言うデータ構造の形で（完了の成否に拘らず）イニシエータのステータスFIFOの示すアドレスにストアする。ステータスブロックの例を、図29に示す。

#### 【0177】

ターゲットは最低8バイト、最大32バイトのステータス情報をストアすることができる。マネージメントORBの場合は、図25のステータスFIFOフィールドにORBの一部として明示的にステータスFIFOアドレスが提供されるので、ターゲットは指定されたアドレスにステータスブロックをストアする。それ以外の場合は、フェッチエージェントのコンテキストから得られるステータスFIFOにステータスブロックをストアする。コマンドブロックORBの場合は、イニシエータはステータスFIFOアドレスをログイン要求の一部として提供する。

#### 【0178】



通常ターゲットは、イニシエータの発するORBに対してステータスFIFOアドレスにステータスブロックを書き込むことによって応答するが、デバイス側に変化が発生し、ロジカルユニットに影響する場合は、イニシエータからの要求が無くても自発的にアンソリシテッドステータス (Unsolicited Status) を返すこともできる。この場合のステータスFIFOアドレスは、イニシエータからのログイン要求の際にイニシエータから提供されるステータスFIFOアドレスである。

#### 【0179】

SBP-2におけるイニシエータとターゲットの動作を、図30の動作モデルを用いて説明する。

#### 【0180】

SBP-2の動作は、イニシエータが、ターゲットに対してログイン要求のためのマネージメントORB (ログインORB) を発行することから始まる。ターゲットは、イニシエータからの要求に対してログインレスポンスで応える。

#### 【0181】

ログイン要求が受け入れられると、ターゲットからはログインレスポンスとしてコマンドブロックエージェントレジスタの先頭アドレスがかえされる。

#### 【0182】

ログイン要求が受け入れられると、ターゲットのマネージメントエージェントは、イニシエータからのその後のタスクマネージメント要求を受け付ける。イニシエータは、タスクマネージメントORBを発行して、タスクの実行に必要な情報のやり取りをターゲットとの間で行う。ターゲットはイニシエータの発するORBに対してステータスFIFOにステータスブロックを書き込むことによって応答するが、デバイス側に変化が発生し、ロジカルユニットに影響する場合は、イニシエータからの要求が無くても自主的にアンソリシテッドステータスを返すこともできることは前述の通りである。

#### 【0183】

タスクマネージメントに関するやり取りに続いて、イニシエータは必要なコマンドブロックORB (list) を自分のメモリ領域に形成する。そして、図31の

ように、ターゲットのコマンドブロックエージェントレジスタのORBポインタにORBの先頭アドレスを書きこむか又はコマンドブロックエージェントレジスタのドアベルレジスタを叩いて、ターゲットに対して通信すべきORBがイニシエータにあることを知らせる。

#### 【0184】

これに応じて、ターゲットは、図32のように、ORBポインタに書かれたORBの先頭アドレス情報をもとにイニシエータのメモリにアクセスし、ORBを順次処理する。

#### 【0185】

ところで、タスクの実行モデルには、オーダードモデルとアンオーダードモデルがある。オーダードモデルでは、ORBはリストの順番に沿って行われ、ターゲットの完了ステータスも順番に返される。アンオーダードモデルでは、ORBの実行順位に制約は無いが、どの順位でも最終的に同じ実行結果が得られるようにイニシエータが責任を持たなければならない。

#### 【0186】

イニシエータからターゲットへのデータ転送はターゲットからシステムメモリへのリードトランザクションによって行われ、一方ターゲットからイニシエータへのデータ転送は、システムメモリへのライトトランザクションによって行われる。即ちデータバッファの転送は方向によらずターゲットが主導する。逆に言えば、ターゲットは自分に都合の良いペースでシステムメモリからのデータを読み出すことができる訳である。イニシエータはターゲットがORBを実行中でも、リストの最後のORBのネクストアドレスを次のORBのアドレスに書き換え、ターゲットのドアベルレジスタを再び叩いてターゲットに変更を知らせることにより、リンクリストにORBを追加することができる。ターゲットは、イニシエータのステータスFIFOのアドレスに完了ステータス（ステータスブロック）を返す。イニシエータは完了ステータス（ステータスブロック）が返されたのを見て、その対象ORBのターゲットによる実行が完了したことを知る。完了したORBは、（ネクストORBフィールドがnullでなければ）タスクセットのリンクリストから外すことができる、イニシエータはその空いたリソースを利用して、

必要なら次のコマンドをコマンドリンクリストの最後に追加しても良い。

【0187】

このようにしてORBが実行されタスクが実行される。

【0188】

タスクが終了し、アクセスをつづける必要が無い場合は、イニシエータはログアウトORBを発行し、ターゲットが応えてログアウトが完了する。

【0189】

(15) SBP-3の概要

SBP-3 (Serial Bus Protocol3) は、SBP-2を拡張し機能を追加することによりSBP-2において効率が悪かった点、欠いていた機能を補充する目的で2000年後半から規格化作業が行なわれている。

【0190】

SBP-3において拡張された代表的な機能を挙げると以下の通りになる。

1. アイソクロナスデータ転送機能

2. デバイスハンドルによるイニシエータ/ターゲット特定にとるノードID

非依存機能

3. 1つのORB内の双方向データ転送機能

ここでは上記機能のうち3について説明する。

【0191】

SBP-3はSBP-2と下位互換性を持っており、その基本的なデータ転送シーケンスはSBP-2の概要で説明した通りである。すなわち、イニシエータからターゲットへのデータ転送はターゲットからシステムメモリへのリードトランザクションによって行われ、一方ターゲットからイニシエータへのデータ転送は、システムメモリへのライトトランザクションによって行われる。ターゲットは自分に都合の良いペースでORBを読み出し、ORBの内容をデコードすることによって転送動作の種別情報を知る。ターゲットはORBに対応する転送動作がイニシエータからターゲットへのデータ転送か、ターゲットからイニシエータへのデータ転送なのか、そしてその転送動作がイニシエータ内のどのシステムメモリ領域に対して行なわれるか、をデコードし、該当する転送動作を行なう。ま

たターゲットはそのORBで指定された転送動作を完了した際には、イニシエータのステータスFIFOのアドレスに完了ステータス（ステータスブロック）を返す。

#### 【0192】

SBP-3ではコマンドブロックORB、つまり、データ転送コマンド、デバイス制御コマンド等のコマンドを内包するORBを2種類定義している。ひとつはコマンドブロックORBが有するリクエストフォーマットフィールドの値が0のものであり、SBP-2で定義されたコマンドブロックORBと同一のものである。「(14) SBP-2の概要」で説明したように、ORBが参照するデータバッファ又はページテーブルのアドレス及びサイズを示すデータディスクリプタ、データサイズフィールド、次のコマンドブロックORBのアドレスを示すネクストORBフィールド、そしてデータバッファに対するデータ転送方向を示すディレクションフィールドに代表されるデータ転送に関わるパラメータを指定するフィールドからなる。

#### 【0193】

SBP-3で新規に定義されたコマンドブロックORBはリクエストフォーマットフィールドの値が1となっており従来のORBと区別がつけられる。

#### 【0194】

このORBの特徴は1つのORBから2つのデータバッファが参照されるということである。

#### 【0195】

データバッファ又はページテーブルのアドレス及びサイズを示すデータディスクリプタ、データサイズフィールド、ディレクションフィールド等データバッファに関するフィールドがそれぞれ2組用意され、これによりORBから2つのデータバッファの参照が可能となる。

#### 【0196】

このORBを使用し、一つのデータバッファはターゲットへのライト用バッファ、もう一方のデータバッファはターゲットからのリード用バッファとして使うことにより双方向ORBとして活用することができる。

**【0197】**

イニシエータは必要なコマンドブロックORBリストを自分のメモリ領域に形成し、上記のように双方向ORBをアペンドしてターゲットのコマンドブロックエージェントレジスタのORBポインタにORBリストの先頭アドレスを書きこむか又はコマンドブロックエージェントレジスタのドアベルレジスタを叩いて、ターゲットに対して通信すべきORBがイニシエータにあることを知らせる。ターゲットは、ORBポインタに書かれたORBの先頭アドレス情報をもとにイニシエータのメモリにアクセスし、ORBをフェッチし、順次処理する。

**【0198】**

双方向ORBをフェッチしたターゲットは指定された2つのデータバッファそれぞれに対してデータ転送処理を行なう。片側のバッファに対応するディレクションフィールドは0となっていて、ターゲットはデータディスクリプタで示されたシステムメモリ上のデータバッファにアクセスし、そこからライトデータを読みこむ。またもう片方ののバッファに対応するディレクションフィールドは1となっていてターゲットはデータディスクリプタで示されたシステムメモリ上のデータバッファにアクセスし、そこへコマンドの要求するデータを書きこむ。

**【0199】**

両方のデータバッファアクセスが完了するとターゲットはイニシエータのステータスFIFOのアドレスに完了ステータス（ステータスブロック）を返し、そのORBの実行完了を通知する。

**【0200】**

図33は、SBP-3でのコマンドブロックORBの動作を示す図である。図32で示したSBP-2と比較すると、SBP-3では1つのORBに対してデータバッファアクセスを示す線が2本になっており、SBP-2が1つのデータバッファへのアクセスしかできなかった点が変更されている。2つのデータバッファに独立してアクセスできることにより、イニシエータからターゲットへの2つのデータチャネルを1つのORBに当てはめることができる。データバッファへのデータの流れる方向はそれぞれのデータバッファで規定できる。つまり、2つのデータチャネルをイニシエータからターゲットへのデータ送信に用いること

もできるし、一方をイニシエータからターゲットへのデータ送信、他方をターゲットからイニシエータへのデータ送信に用いることもできる。また、もちろん、ターゲットからイニシエータへのデータ送信に2つのデータチャネルを用いることもできる。更に、ホストとプリンタで使うような場合、一方をホストからプリンタへの印字データのために使い、他方をプリンタからホストへのステータス情報を流すために使うことも可能となる。

#### 【0201】

図34にSBP-3の2つのデータバッファを保持する場合のコマンドブロックORBの詳細を示している。SBP-2と異なり、2組のデータバッファを扱えるようにするために、2つのデータディスクリプタと2つのバッファ情報フィールドが用意されている。図中"d"で示したフィールドにそれぞれのデータバッファの方向を指定することができる。その内容はSBP-2と同様であって、0であればイニシエータからターゲット、1であればターゲットからイニシエータへの方向を示している。

#### 【0202】

〈各ノードの構成〉

まず、IEEE1394バスに接続される各ノードの1394シリアルバスインターフェース部の構成を説明する。

#### 【0203】

図35は1394インターフェースブロックの基本構成ブロック図である。

#### 【0204】

図中3502は1394シリアルバスを直接ドライブするフィジカルレイヤー制御IC(PHYIC)であり、前述の〈IEEE1394の技術の概要〉におけるフィジカルレイヤの機能を実現する。主な機能としては、バスイニシャル化とアービトレーション、送信データ符号のエンコード/デコード、ケーブル通電状態の監視ならびに負荷終端用電源の供給(アクティブ接続認識用)、リンクレイヤICとのインターフェースである。

#### 【0205】

3501はデバイス本体とのインターフェースを行い、PHYICのデータ転

送をコントロールするリンクレイヤー制御 IC (LINK IC) であり、前述の〈IEEE 1394 の技術の概要〉におけるリンクレイヤの機能を実現する。本 IC が備える主な機能としては PHY IC を介する送信／受信データを一時格納する送受信 FIFO、送信データの packets 化機能、PHY IC が受信データが本ノードアドレス、またはアイソクロナス転送データの場合は割り当てられたチャンネル向けのものであるかの判定機能、またそのデータのエラーチェックを行うレシーバー機能、そしてデバイス本体とのインターフェースを行う機能がある。

#### 【0206】

図中 3504 はリンクレイヤ IC、PHY IC をはじめとする 1394 インターフェース部をコントロールする CPU であり、3505 は同インターフェース部のコントロール用プログラムが格納されている ROM である。

#### 【0207】

3506 は RAM であり、送受信データを蓄えるデータバッファをはじめ、制御用ワークエリア、1394 アドレスにマッピングされた各種レジスタのデータ領域に使用されている。

#### 【0208】

図中 3503 はコンフィギュレーション ROM であり、各機器固有の識別、通信条件等が格納されている。本 ROM のデータフォーマットは〈IEEE 1394 の技術の概要〉で説明したように IEEE 1212 並びに IEEE 1394 規格で定められたフォーマットに準じている。

#### 【0209】

各ノードは図 36 に示すような一般形式のコンフィギュレーション ROM を装備しており、各デバイスのソフトウェアユニット情報はユニットディレクトリに、ノード固有の情報はノードディペンデントインフォディレクトリに保存されている。

#### 【0210】

本実施の形態のプリンタでの場合、通信プロトコルとして SBP-3 をサポートするためにユニットディレクトリに SBP-3 を識別する ID が装備されてい

る。

#### 【0211】

〈IEEE 1394の技術の概要〉で説明したように1394シリアルバスのアドレス設定のうち、最後の28ビットはシリアルバスに接続される他のデバイスからアクセス可能な、各機器の固有データの領域として確保されている。図37はこの28ビットのアドレス空間を表した図である。

#### 【0212】

図中0000番地から0200番地の領域にはCSRコアレジスタ群が配置されている。

#### 【0213】

これらレジスタはCSRアーキテクチャで定められたノード管理のための基本的な機能として存在している。

#### 【0214】

0200番地から0400番地の領域は、CSRアーキテクチャにより、シリアルバスに関するレジスタが格納される領域として定義されている。この領域の詳しい構成を図38に示す。図38においては、シリアルバスに関するレジスタとして、0200～0230番地のレジスタが定義されておりデータ転送の同期、電源供給、バスリソース管理等に使用されるレジスタが配置されている。なお、コンフィギュレーションROMは400番地から800番地の領域に配置されている。

#### 【0215】

0800番地から1000番地までの領域には、現在の1394バスのトポロジ情報、またノード間の転送スピードに関する情報が格納されている。

#### 【0216】

1000番地以降の領域はユニット空間と呼ばれ、各デバイス固有の動作に関連するレジスタが配置されている。この領域には各デバイスがサポートする上位プロトコルで規定されたレジスタ群とデータ転送用メモリマップドバッファ領域、また各機器固有のレジスタが配置される。

#### 【0217】



〈本実施形態に係る情報処理システム〉

図1は、本実施形態の情報処理システムを示す図であり、IEEE1394で接続されたホストコンピュータ（以下ホスト）1aとプリンタ1bを表している。

【0218】

ホスト1aとプリンタ1bの通信はIEEE1394インターフェース上で使われる代表的なデータ転送プロトコルであるSBP（Serial Bus Protocol）-3プロトコルを使って行なわれ、ホスト-プリンタ間のデータ転送を行なうための通信チャネルとしてLUN0が予め定められている。

【0219】

図1中、ホスト1aはSBP-3プロトコルを用いたイニシエータとしてSBP-3のターゲットであるプリンタ装置1bとの間で通信を行うシステムである。またこの通信システムの場合、SBP-3プロトコルの上位コマンドセットとして予め規定されたプリンタ制御用通信コマンドプロトコルに基づいてホスト-プリンタ間通信が行なわれ、コマンドに従った処理が行なわれる。

【0220】

プリンタ1bはホスト1aから送られたプリンタコントロール、並びにプリントデータを受信し処理する機能と、プリンタ1bへのコントロールに対応して現在のプリンタステータスデータをホストに送信する機能とをそれぞれ提供する複数のデータ転送チャネルCH1、CH2を備える。またこれらのデータ転送チャネル用の通信セッション確立、切断といったマネージメントリクエストを処理するマネージメントエージェントMAを装備している。

【0221】

各データ転送チャネルCH1、CH2はSBP-3プロトコルで定義されたデュアルデータディスクリプタを備えたORBの活用によりデータ転送が行なわれる。それぞれのチャネルは二つのデータディスクリプタで指定されるデータバッファによりデータ転送を行なう。フェッチエージェントFAによりフェッチされるORBはコマンドブロック・エージェントCAにより実行状態が管理され、ORBで指定される2つのデータバッファに対する所定の処理はエクセキューショ

ン・エージェント E A 0、E A 1によりライト、またはリード処理される。フェッチエージェントは S B P - 2 / 3 のオーダードモデルに従いイニシエータから発行される O R B をシーケンシャルに処理することにより上記所定の処理を実行させる仕組みとなっている。

#### 【 0 2 2 2 】

本実施形態では、プリンタ制御用通信コマンドプロトコルとして S B P - 3 を使用することにより、O R B 中二つのデータディスクリプタで指定されるデータバッファをそれぞれデータ転送チャネルとしてアロケートし、一方のデータ転送チャネルにはプリントデータを、もう一方のデータ転送チャネルにはプリンタ制御コマンドの転送が行なわれるモデル仕様として定義されている。

#### 【 0 2 2 3 】

通常、S B P - 3 ではターゲットは O R B で指定されたデータバッファに対する処理が完了した際に、それをイニシエータに通知するための完了通知としてイニシエータのステータス F I F O のアドレスに完了ステータス（ステータスブロック）を発行する。しかし、O R B において 2 つのデータディスクリプタフィールドによって 2 つのデータバッファが指定されている場合に、両方のデータバッファ処理が完了した時点で完了ステータス（ステータスブロック）を発行するのでは、片方のデータバッファに対するアクセスが正常終了しても、もう片方のデータバッファアクセスが完了しないとその O R B に対する完了通知を S B P - 3 のイニシエータに対して通知できず、L U N としてのデータ転送が滞ってしまう。このため、何らかの理由によりどちらかのデータ転送チャネルにおいてデータの送受信ができなくなった場合、またどちらかのデータ転送チャネルのデータ転送が他のチャネルよりも遅い場合、正常に転送されているチャネル、速いほうのチャネルが影響を受けてしまう。

#### 【 0 2 2 4 】

そこで、本実施形態では、図 3 9 に示すようなステータスブロックを発行し、データバッファごとのアクセス完了を通知できる構成となっている。

#### 【 0 2 2 5 】

図 3 9 は、本実施の形態に係るプリンタ制御用通信コマンドプロトコルでの、

データバッファ処理が完了した時点で発行する完了ステータス（ステータスブロック）を示す図である。

#### 【0 2 2 6】

図 3 9 に示すように、このステータスブロックには、コマンドセット依存フィールドにデータバッファ毎の完了ステータスをあらわすフィールドとして B 1、B 2 が定義されている。

#### 【0 2 2 7】

その上でターゲットとしてのプリンタ 1 b は、O R B によって指定されたバッファのうち、一方のバッファの転送が完了した時点で、B 1 または B 2 のいずれかに所定の値を代入してステータスブロックを発行する。

#### 【0 2 2 8】

これにより、それぞれのバッファにアロケートされているデータ転送チャンネルにおいて、他方チャンネルのデータ転送処理の進行具合に関わらず完了通知を発行することが可能となる。

#### 【0 2 2 9】

また、本実施の形態のコマンドプロトコルは、図 4 0 に示すように、イニシエータが発行する O R B 中、S B P - 3 で定義されているコマンドブロックフィールドにおいて、2 つの I D 格納フィールドを定義している。これは、その O R B によって指定されるデータバッファ単位のデータ内容に付加される I D を格納するフィールドであり、各データバッファ、すなわちデータ転送チャンネル毎に I D フィールドが用意される。データ転送が進行する、すなわち、あるチャンネルにおいてデータ転送中、あるデータバッファの転送処理が正常完了した場合には次の O R B で指定されるデータバッファの I D は 1 インクリメントされた値となる。

#### 【0 2 3 0】

何らかの理由により、あるデータバッファがターゲットによって処理されず、再送となった場合には、イニシエータは該当するデータバッファに対して同じ I D を付加した O R B を用意する。

#### 【0 2 3 1】

この機能によりターゲットは I D フィールドを確認することで、アペンドされた O R B で指定されるデータバッファから新規にデータが転送されるのか、前回の O R B リスト中で既に指定されたデータバッファからデータが再送されるのか、を知ることが可能となる。

#### 【 0 2 3 2 】

以上のシステムにおけるデータ転送を説明する。

#### 【 0 2 3 3 】

イニシエータであるホスト 1 はプリントジョブを実行するために上記ターゲット 2 に対してプリンタの論理ユニット L U N 0 への接続を行なう。

#### 【 0 2 3 4 】

プリンタはホストから通信開始の要求、すなわちログイン要求があると、マネージメントエージェント M A はフェッチエージェント F A へアクセスするためのエントリポイントとなるベースアドレスを含んだ応答をログインレスポンスとしてイニシエータへ返送することによりイニシエータ 1 に対して通信を許可する。

#### 【 0 2 3 5 】

ホストは L U N 0 にログイン要求を発行し、通信が許可されると通信を開始する。

#### 【 0 2 3 6 】

ホストはプリンタに対して L U N 0 へのログイン、そしてその L U N 上でデュアルディスクリプタを備えた O R B の使用により 2 つのデータ転送チャネルを確保する。ひとつのデータチャネル C H 1 はプリンタの制御コマンドの送信、並びにプリンタステータス受信用の双方向チャネル、もう一方のチャネル C H 2 は印刷データの送信を行なうためのチャネルとして用いる。

#### 【 0 2 3 7 】

イニシエータは印刷動作をプリンタに対して指示する場合、プリント動作を指定するプリントアプリケーションコマンドの発行、プリンタのステータスを問い合わせるアプリケーションコマンド等プリンタ制御に必要なコマンドを C H 1 のデータバッファに、実際に印刷する印刷データを C H 2 のデータバッファに展開、それぞれのチャネルを参照する O R B リスト（ここでは O R B 3 つのリスト）

をメモリ上に用意、ORBをアペンドした状態でプリンタのフェッチエージェントに対してORBのフェッチをトリガする。

#### 【0238】

ORBポインタのアクセスによりフェッチの起動要求を受けたターゲットはその要求に含まれるポインタ情報を元に、該当ORBを、IEEE1394のリードトランズアクションを使いフェッチする。ORBをデコードし、リクエストフォーマットフィールド（図中rq\_fmt）の値によりORBがデュアルディスクリプタタイプであることを認識し、それぞれのディスクリプタに対して付加されたディレクションビットの値に応じてデータバッファへのアクセスを行う。CH1の場合、プリントコマンド送信時にはプリンタへのライトコマンド、すなわちホストからプリンタへのデータ転送となり、エグゼキューションエージェントはデータバッファを読み出す処理を行なう。

#### 【0239】

一方で、プリンタステータスの取得時にはプリンタへのリードコマンド、すなわちプリンタからホストへのデータ転送となりエグゼキューションエージェントはデータバッファに書き込む処理を行う。CH2の場合、プリントデータの送信に伴うプリンタへのライトコマンド、すなわちホストからプリンタへのデータ転送となりエグゼキューションエージェントはデータバッファを読み出す処理を行う。

#### 【0240】

プリンタは読み込まれたアプリケーションデータに基づいた動作を行なう。印字データ並びに印字指定コマンドの送信、プリンタデバイスの初期化といったホストからプリンタへの制御データ送信、そしてプリンタステータスのホストへの送信がそれに相当する。

#### 【0241】

以上の動作によりCH1、CH2のデータ転送は実行されるが、プリンタの動作状態によって、CH1、CH2のデータ転送が同じペースで実行されないケースがある。すなわち、あるORBに対してCH1のデータバッファ転送は完了しているにも関わらず、同じCH2のデータバッファ処理が何らかの理由により滞

る場合がある。本実施の形態におけるこの場合の動作を説明する。

#### 【0 2 4 2】

この場合、ターゲットプリンタはORBによって指定されたバッファのうち、CH 1 側バッファの転送が完了した時点で、本実施の形態のプリンタ制御用通信コマンドプロトコルにおいて図 3 9 に示すステータスブロックで定義した、該当バッファの完了を示すフィールド（図中B 1 またはB 2、ここではCH 1 側が完了しているのでB 1）に、完了を示す所定の値を代入して完了ステータスを発行する。

#### 【0 2 4 3】

これにより、イニシエータは、このORBにおいてCH 1 のデータバッファ処理は完了したものの、CH 2 のデータバッファ処理は完了していないことがわかる。一方、ターゲットプリンタはCH 2 のデータ転送処理の進行具合に関わらずCH 1 に対する完了通知を発行することできるので、CH 2 のデータ転送状況と独立してCH 1 のデータ転送を進行させることが可能となる。

#### 【0 2 4 4】

ORBに対する完了ステータスを発行したターゲットプリンタは、ORBリストにORBが残っている場合、次のORBを同様に処理していく。CH 2 のデータ転送が未だ滞っている場合は上記と同様にB 1 にのみ完了を示す所定の値を代入してステータスブロックをステータスF I F Oに発行していく。

#### 【0 2 4 5】

ターゲットプリンタがORBリストを完了し、最後のORBに対するステータスブロックを発行すると、イニシエータは次のORBリストを準備する。

#### 【0 2 4 6】

この際、イニシエータは前回のORBリストにおけるステータスブロックにおいて未完了となっているチャンネルの該当データバッファを再アペンドする一方で、完了しているチャンネルについては次のデータ転送を行なうべく新たなデータバッファをアペンドする。

#### 【0 2 4 7】

これによりデータ転送が進んでいるチャンネルは次のデータを転送、滞っている

チャンネルには同じデータを再転送することになる。

#### 【0248】

本実施の形態のプリンタ制御用通信コマンドプロトコルにおいてORB中に定義したデータバッファIDフィールドにはアペンドしたデータバッファに対するIDが付加される。

#### 【0249】

従って、図42のように、CH1の場合、新たなデータに対するデータバッファにはインクリメントされたバッファIDが付加される。CH2の場合、前回未完了となったデータバッファ以降のデータ再送となるため、データバッファのIDにはそれぞれ該当するバッファIDが付加される。

#### 【0250】

ORBリストが準備完了となるとプリンタのフェッチエージェントに対してORBのフェッチをトリガする。ターゲットプリンタはORBをフェッチ後、IDフィールドを確認することによりアペンドされたORBで指定されるデータバッファが新規に転送されるデータか、前回のORBリスト中で既に指定されたデータバッファの再送バッファかを知ることが可能となる。ターゲットプリンタはIDによるデータバッファ管理を行なうことにより再度アペンドされたチャンネル（CH2）のデータバッファの内容に対する処理を重複して行なうことなくデータ転送を継続可能となる。一方新たなデータバッファがアペンドされたチャンネル（CH1）はデータ転送が進行する。これによりCH1、CH2は独立したフローコントロールによるデータ転送を行なうことが可能となる。

#### 【0251】

（他の実施形態）

以上、本発明の実施形態について詳述したが、本発明は、複数の機器から構成されるシステムに適用しても良いし、また、一つの機器からなる装置に適用しても良い。また、上記実施形態では通信制御バスとしてIEEE1394を利用した例について説明したが、本発明はこれに限定されるものではなく、USBなどの他の規格のバスを利用しても良い。

#### 【0252】

なお、本発明は、前述した実施形態の機能を実現するソフトウェアのプログラムを、システム或いは装置に直接或いは遠隔から供給し、そのシステム或いは装置のコンピュータが該供給されたプログラムコードを読み出して実行することによっても達成される場合を含む。その場合、プログラムの機能を有していれば、形態は、プログラムである必要はない。

#### 【0253】

従って、本発明の機能処理をコンピュータで実現するために、該コンピュータにインストールされるプログラムコード自体も本発明を実現するものである。つまり、本発明のクレームでは、本発明の機能処理を実現するためのコンピュータプログラム自体も含まれる。

#### 【0254】

その場合、プログラムの機能を有していれば、オブジェクトコード、インタプリタにより実行されるプログラム、OSに供給するスクリプトデータ等、プログラムの形態を問わない。

#### 【0255】

プログラムを供給するための記録媒体としては、例えば、フロッピー（登録商標）ディスク、ハードディスク、光ディスク、光磁気ディスク、MO、CD-ROM、CD-R、CD-RW、磁気テープ、不揮発性のメモ리카ード、ROM、DVD（DVD-ROM、DVD-R）などがある。

#### 【0256】

その他、プログラムの供給方法としては、クライアントコンピュータのブラウザを用いてインターネットのホームページに接続し、該ホームページから本発明のコンピュータプログラムそのもの、もしくは圧縮され自動インストール機能を含むファイルをハードディスク等の記録媒体にダウンロードすることによっても供給できる。また、本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるホームページからダウンロードすることによっても実現可能である。つまり、本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせるWWWサーバも、本発明のクレームに含まれるものである。



**【0257】**

また、本発明のプログラムを暗号化してCD-ROM等の記憶媒体に格納してユーザに配布し、所定の条件をクリアしたユーザに対し、インターネットを介してホームページから暗号化を解く鍵情報をダウンロードさせ、その鍵情報を使用することにより暗号化されたプログラムを実行してコンピュータにインストールさせて実現することも可能である。

**【0258】**

また、コンピュータが、読み出したプログラムを実行することによって、前述した実施形態の機能が実現される他、そのプログラムの指示に基づき、コンピュータ上で稼動しているOSなどが、実際の処理の一部または全部を行ない、その処理によっても前述した実施形態の機能が実現され得る。

**【0259】**

さらに、記録媒体から読み出されたプログラムが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行ない、その処理によっても前述した実施形態の機能が実現される。

**【0260】****【発明の効果】**

本発明によれば、IEEE1394バス等の2つのチャネルを用いて効率的なデータ転送を行うことができる。

**【0261】**

例えば、IEEE1394においてホスト・プリンタ間の通信プロトコルとしてSBP-3を使った場合、SBP-3で規定されたORB単位のスレーブFIPOにディスクリプタ毎のデータバッファ実行完了通知フィールドを設け、データバッファ単位の実行完了通知を行なうと共に、イニシエータは完了しているデータバッファに対してのみ新たなデータバッファのアpendを行ない、更新したデータバッファと更新を行っていないデータバッファを判別することによってORB単位の実行完了を継続させることにより対のデータバッファ（方向）毎に

完全独立なフローコントロールを実現したデータ転送を行なうことを可能とした通信プロトコルを実現することが可能となる。

**【0262】**

これによりSBP-3のひとつの論理ユニット（LUN）においてデュアルディスクリプタを使って2チャネル通信を行なう場合、何らかの理由により片方のデータ通信チャネルが通信不可能状態に陥った場合においても、もう片方のチャネルは滞ることなくデータ転送を行なうことが可能となる。

**【図面の簡単な説明】**

**【図1】**

本発明のホスト・プリンタ通信システムの構成を示した図である。

**【図2】**

1394シリアルバスのネットワークの構成を示した図である。

**【図3】**

本発明の1394シリアルバスの構成要素を示した図である。

**【図4】**

本発明の1394シリアルバスのリンク・レイヤ提供可能なサービスを示す図である。

**【図5】**

本発明の1394シリアルバスのトランザクション・レイヤ提供可能なサービスを示す図である。

**【図6】**

1394インタフェースにおけるアドレス空間を説明する図である

**【図7】**

1394インタフェースにおけるCSRコア・レジスタに格納される情報のアドレス及び機能を示す図である。

**【図8】**

1394インタフェースにおけるシリアルバス・レジスタに格納される情報のアドレス及び機能を示す図である。

**【図9】**

1394 インタフェースにおける最小形式のコンフィグレーションROMを示す図である。

【図10】

1394 インタフェースにおける一般形式のコンフィグレーションROMを示す図である。

【図11】

1394 インタフェースにおけるシリアルバス装置レジスタに格納される情報のアドレス及び機能を示す図である。

【図12】

IEEE 1394 規格に準拠した通信ケーブルの断面図である。

【図13】

DS-Link符号化方式を説明する図である。

【図14】

バスリセットの開始からノードIDの割り当て処理までの基本的なシーケンスを示した図である。

【図15】

バスリセットの開始からノードIDの割り当て処理までの基本的なシーケンスを示した図である。

【図16】

バスリセットの開始からノードIDの割り当て処理までの基本的なシーケンスを示した図である。

【図17】

図15に示したステップS1505の処理（即ち、自動的に各ノードのノードIDを割り当てる処理）を詳細に説明するフローチャートである。

【図18】

1394 インターフェースにおけるセルフIDパケットの構成を示した図である。

【図19】

1394 ネットワークにおけるアービトレーションを説明する図である。

**【図 20】**

1 通信サイクルにおいてアイソクロナス転送モードとアシンクロナス転送モードとを混在させた場合を説明する図である。

**【図 21】**

アイソクロナス転送モードに基づいて転送される通信パケットのフォーマットを示した図である。

**【図 22】**

本発明のアシンクロナス転送のパケットフォーマットを示した図である。

**【図 23】**

S B P - 2 における O R B 種別を示した図である。

**【図 24】**

S B P - 2 におけるコマンドブロック O R B のフォーマットを示した図である。

**【図 25】**

S B P - 2 におけるマネージメント O R B のフォーマットを示した図である。

**【図 26】**

S B P - 2 におけるコマンドブロック O R B のリンクリストを示した図である。

**【図 27】**

S B P - 2 におけるデータバッファの直接アクセスを示した図である。

**【図 28】**

S B P - 2 におけるページテーブルの使用を示した図である。

**【図 29】**

S B P - 2 におけるステータス F I F O のフォーマットを示した図である。

**【図 30】**

S B P - 2 におけるログイン動作を示した図である。

**【図 31】**

S B P - 2 にタスク実行の最初の動作を示した図である。

**【図 32】**

SBP-2におけるコマンドORBの動作を示した図である。

【図33】

SBP-3におけるコマンドORBの動作を示した図である。

【図34】

SBP-3におけるコマンドブロックORBのフォーマットを示した図である。

【図35】

本実施の形態におけるプリンタ・ホストのIEEEインターフェース部のブロック図である。

【図36】

本実施の形態におけるプリンタのコンフィグレーションROMを示す図である。

【図37】

本実施の形態におけるプリンタの1394アドレス空間を示す図である。

【図38】

本実施の形態におけるプリンタのコアCSRレジスタを示す図である。

【図39】

本実施の形態におけるプリンタ・ホスト間で使用されるプリンタ制御用通信コマンドプロトコルで定義されたステータスFIFOのフォーマットを示した図である。

【図40】

本実施の形態におけるプリンタ・ホスト間で使用されるプリンタ制御用通信コマンドプロトコルで定義されたコマンドORBのフォーマットを示した図である。

【図41】

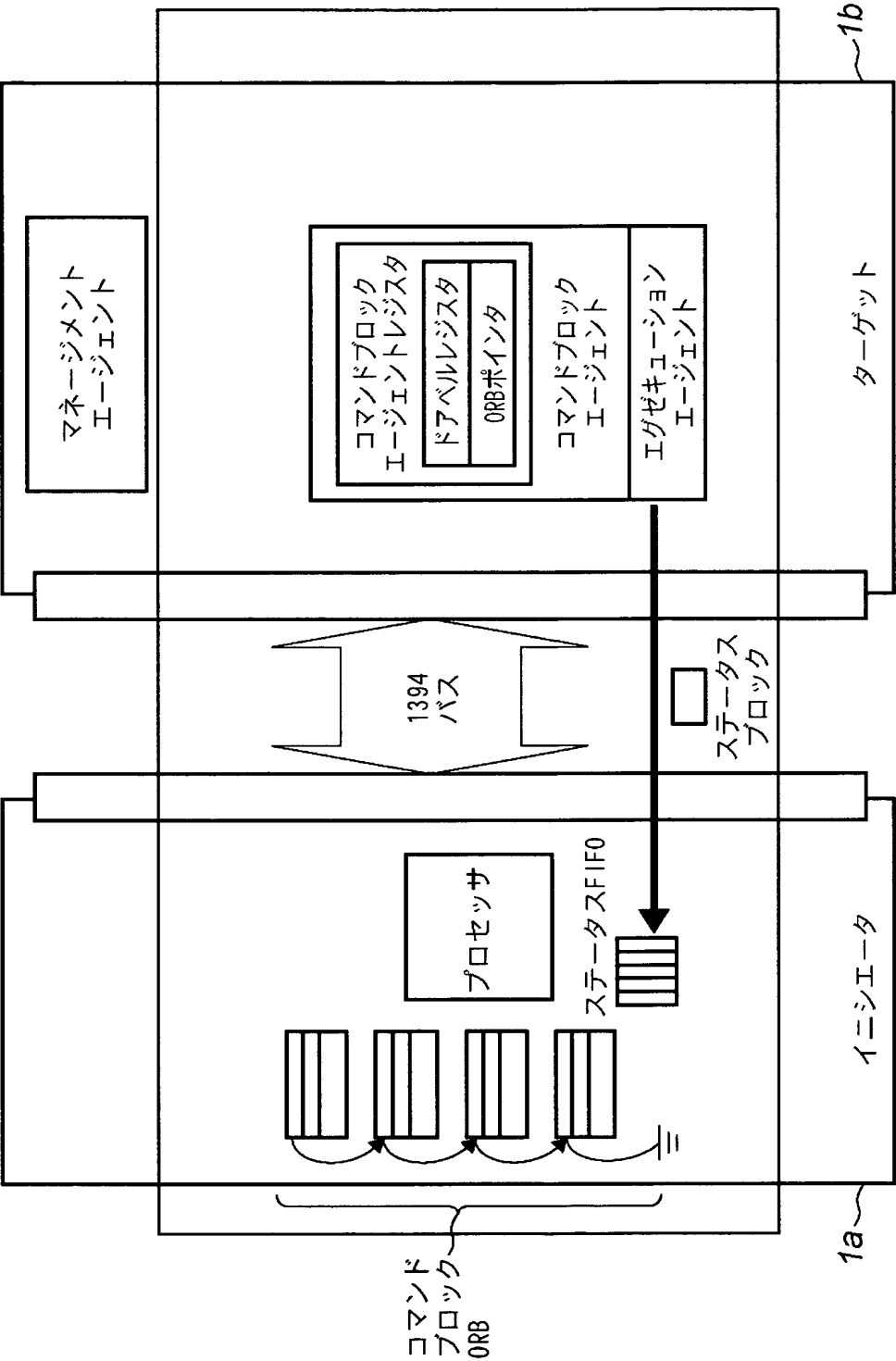
本実施の形態におけるプリンタ・ホスト間で使用されるプリンタ制御用通信コマンドプロトコルの実施の形態における動きのうち、最初のORBリストの処理を示した図である。

【図42】

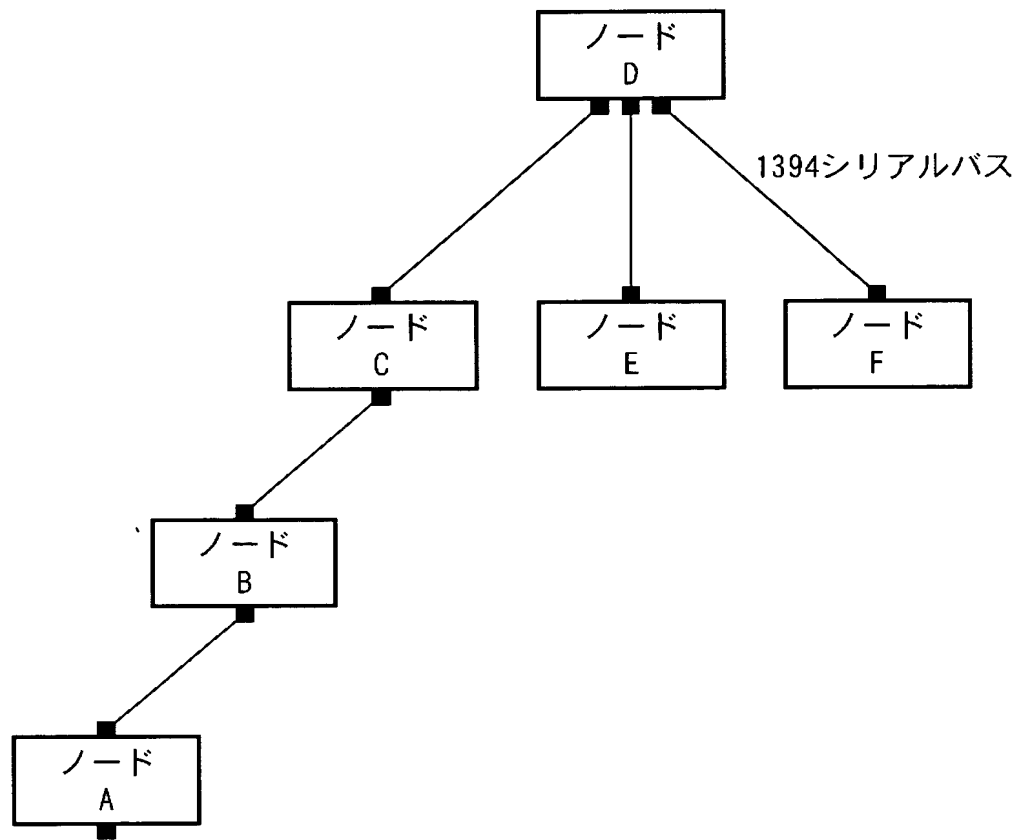
本実施の形態におけるプリンタ・ホスト間で使用されるプリンタ制御用通信コマンドプロトコルの実施の形態における動きのうち、2回目のORBリストの処理を示した図である。

【書類名】 図面

【図 1】

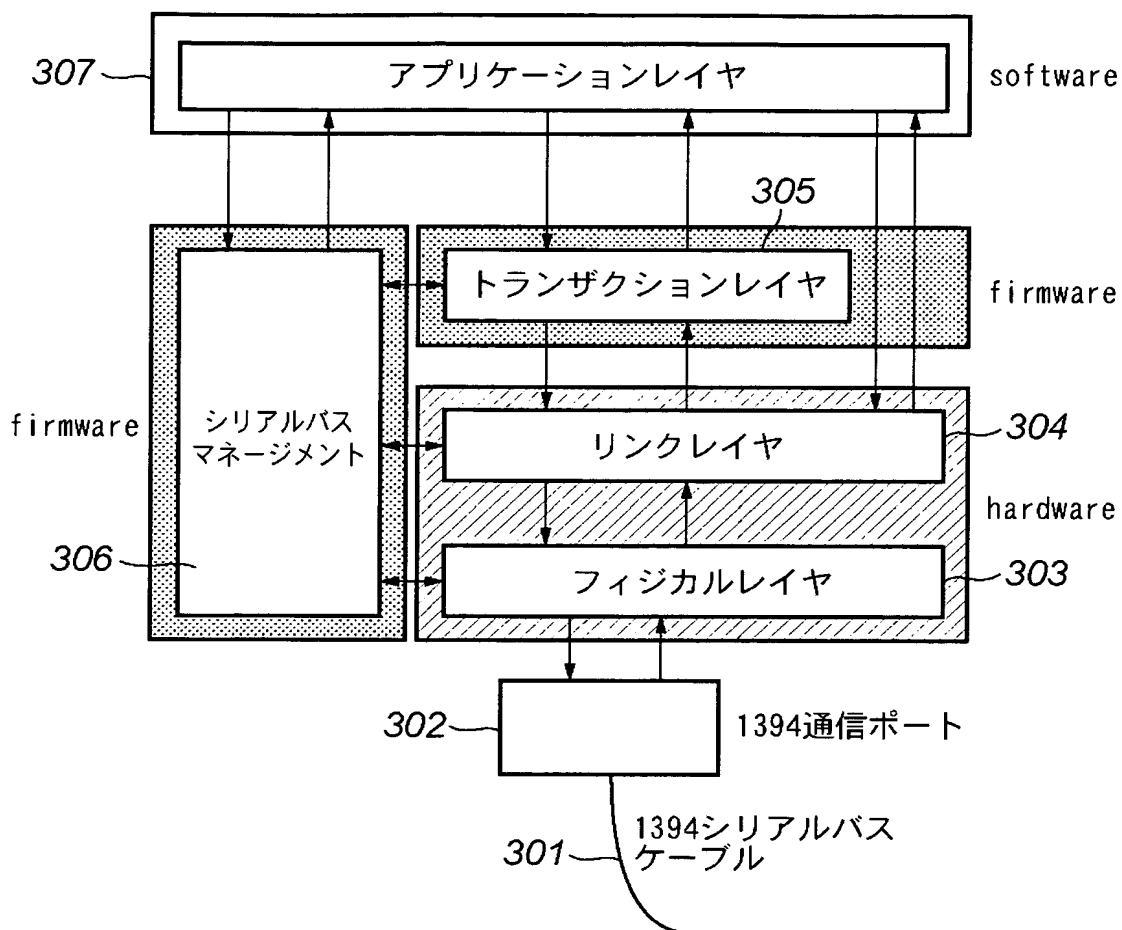


【図 2】

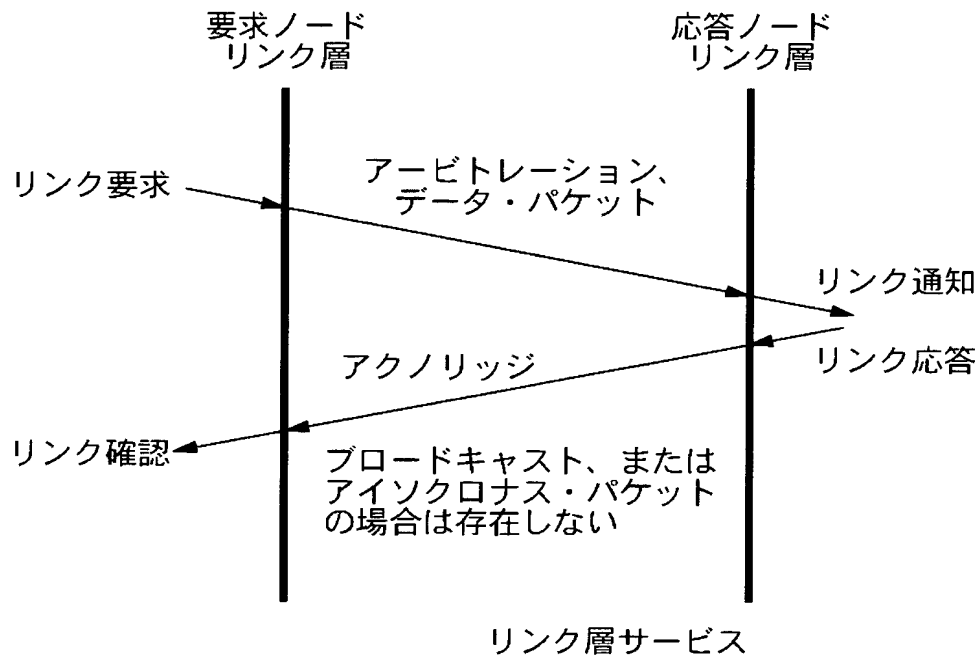




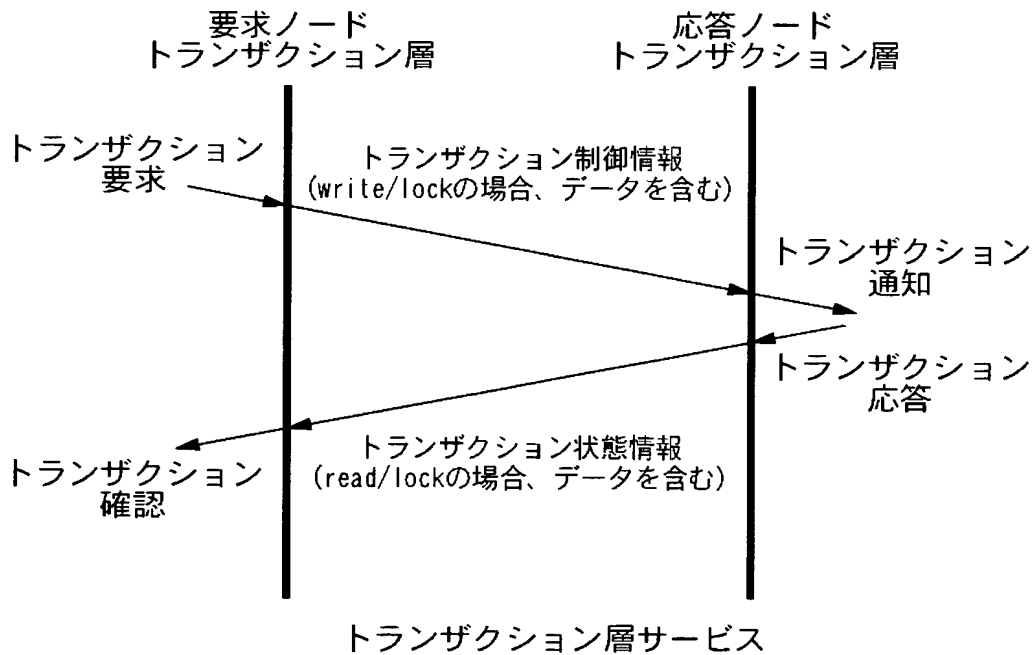
【図 3】



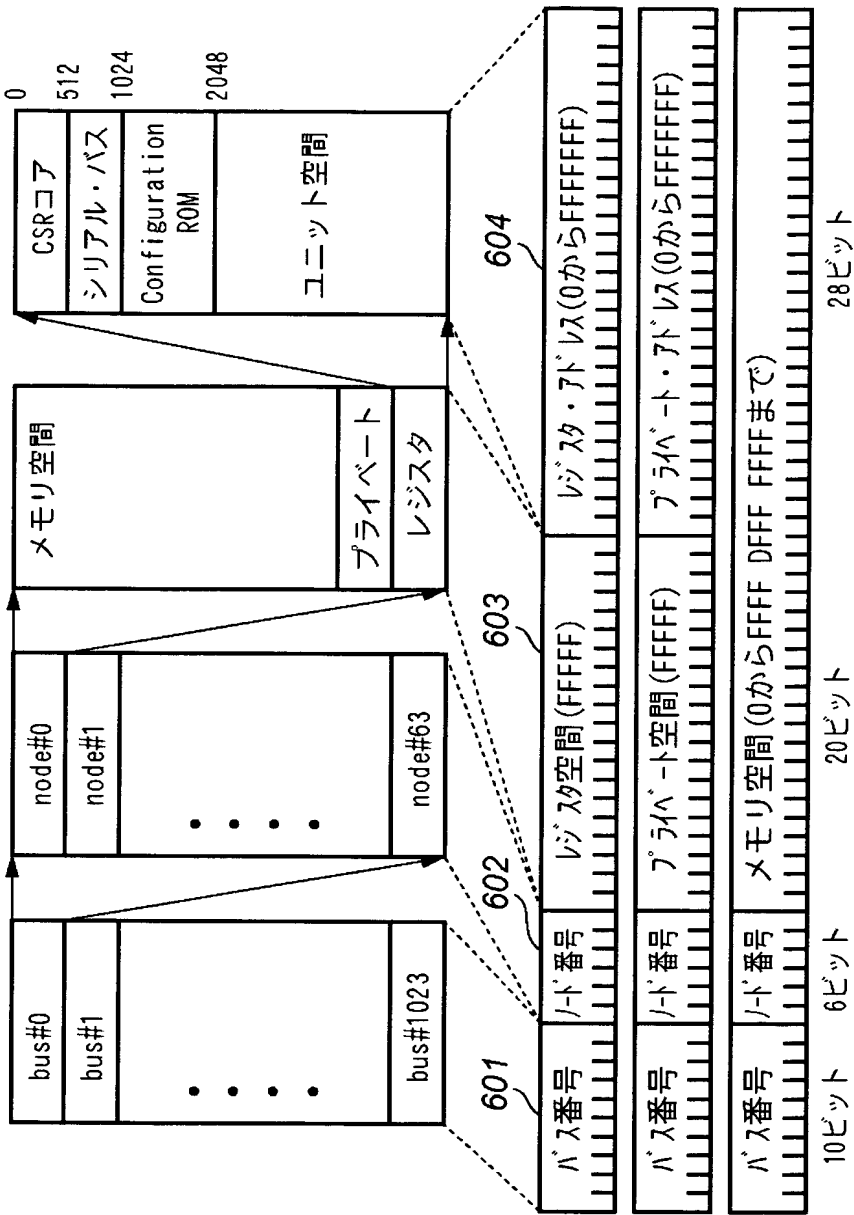
【図 4】



【図 5】



【図 6】



【図 7】

## CSRコア・レジスタ

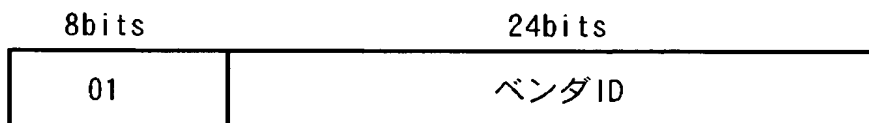
オフセット (16進数)	レジスタ名称	機能
000	STATE_CLEAR	状態と制御の情報
004	STATE_SET	STATE_CLEARの書き込み 可否を示す情報
008	NODE_IDS	バスID+ノードID
00C	RESET_START	この領域に対する書き込みで バスをリセット
010～014	INDIRECT_ADDRESS, INDIRECT_DATA	1Kより大きいROMを アクセスするためのレジスタ
018～01C	SPLIT_TIMEOUT	スプリット・トランザクション のタイムアウトを検出する タイマの値
020～02C	ARGUMENT, TEST_START, TEST_STATUS	診断用のレジスタ
030～04C	UNITS_BASE, UNITS_BOUND, MEMORY_BASE, MEMORY_BOUND	IEEE1394では、実装しない
050～054	INTERRUPT_TARGET, INTERRUPT_MASK	割り込み通知レジスタ
058～07C	CLOCK_VALUE, CLOCK_TICK_PERIOD, CLOCK_STOROB_E_ARRIVED, CLOCK_INFO	IEEE1394では、実装しない
080～0FC	MESSAGE_REQUEST, MESSAGE_RESPONSE	メッセージ通知レジスタ
100～17C		予約
180～1FC	ERROR_LOG_BUFFER	IEEE1394用に予約

【図 8】

## シリアル・バス・レジスタ

オフセット (16進数)	レジスタ名称	機能
200	CYCLE_TIME	アイソクロナス転送のためのカウンタ
204	BUS_TIME	時間を同期するためのレジスタ
208	POWER_FAIL_IMMINENT	電源供給に関するレジスタ
20C	POWER_SOURCE	
210	BUSY_TIMEOUT	トランザクション層の再試行を制御
214 } 218		予約
21C	BUS_MANAGER_ID	バス・マネージャのノードID
220	BANDWIDTH_AVAILABLE	アイソクロナス転送の帯域を管理
224 } 228	CHANNELS_AVAILABLE	アイソクロナス転送のチャンネル番号を管理
22C	MAINT_CONTROL	診断用レジスタ
230	MAINT_UTILITY	
234 } 3FC		予約

【図 9】



最小形式のConfiguration ROM

【図 1 0】

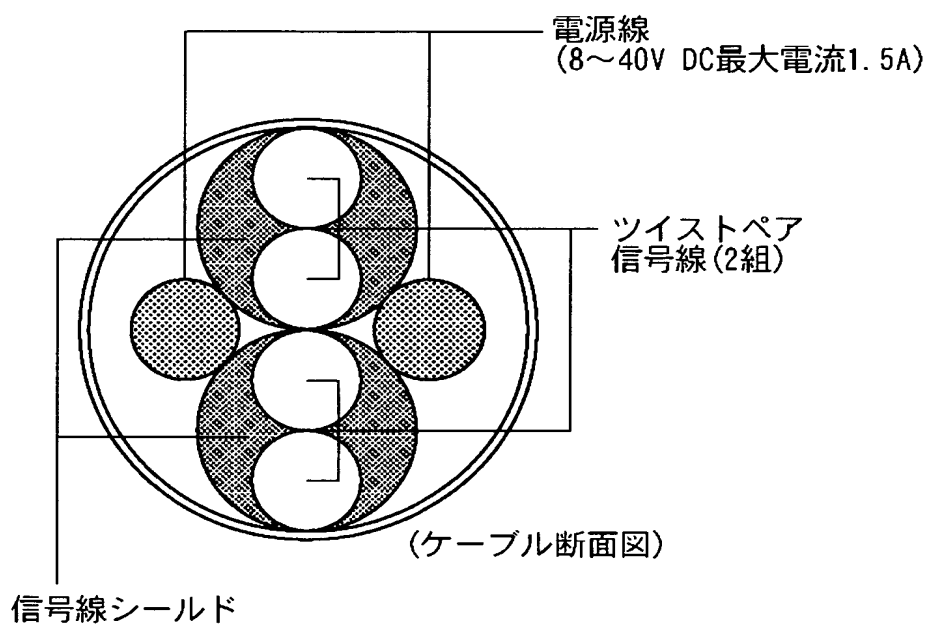
Bus Info Block Length	ROM Length	CRC	
Bus Info Block			~1001
Root Directory			~1002
Node dependent info directory			~1003
Unit directories			~1004
Root & unit leaves			~1005
Vendor dependent information			~1006

【図 1 1】

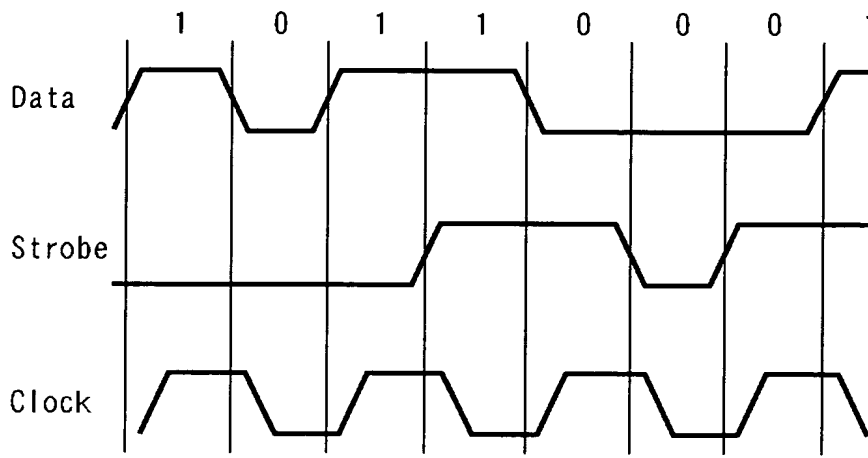
## シリアル・バス装置レジスタ

オフセット (16進数)	レジスタ名称	機能
800 } FFC		予約
1000 } 13FC	TOPOLOGY_MAP	シリアル・バスの構成情報
1400 } 1FFC		予約
2000 } 2FFC	SPEED_MAP	シリアル・バスの転送速度の情報
3000 } FFFC		予約

【図 1 2】

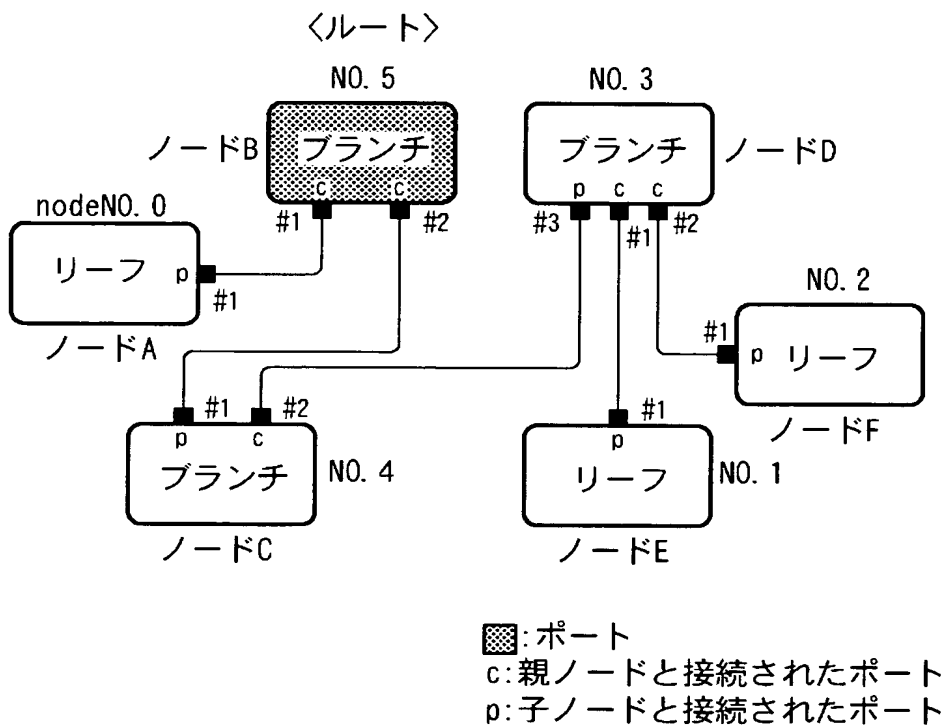


【図 1 3】



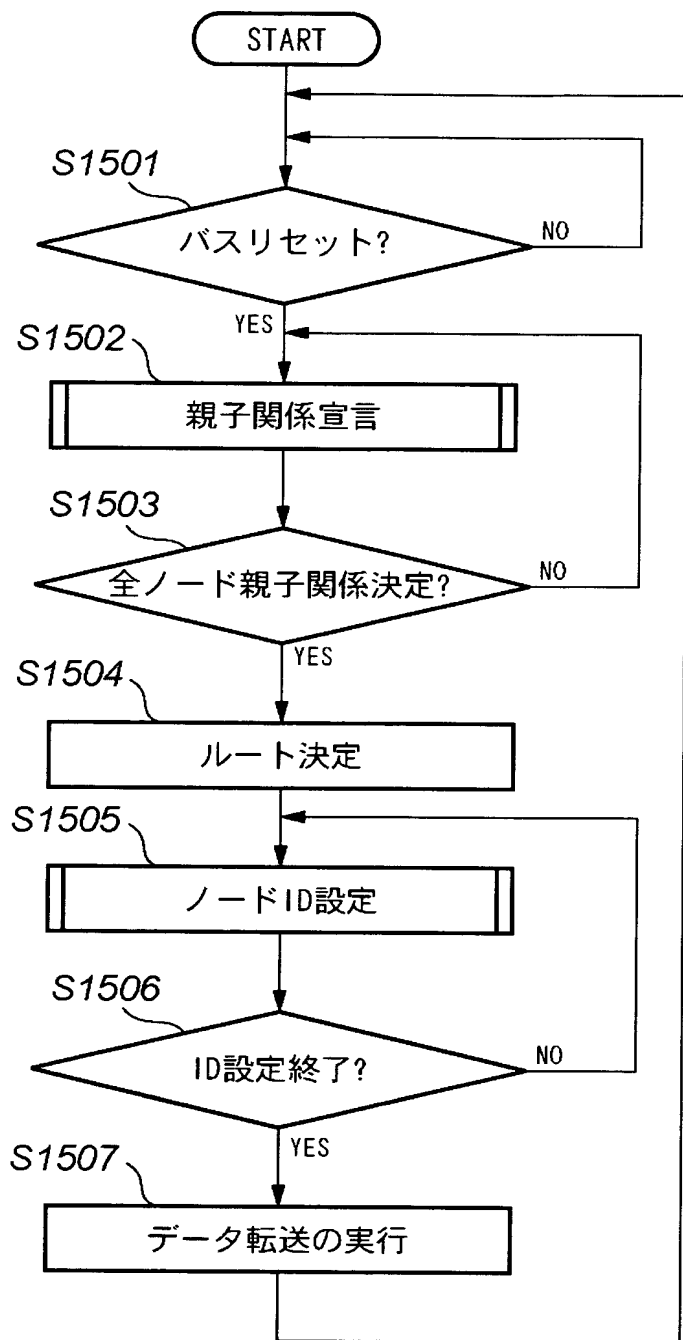
(DataとStrobeの排他的論理和信号)

【図 1 4】

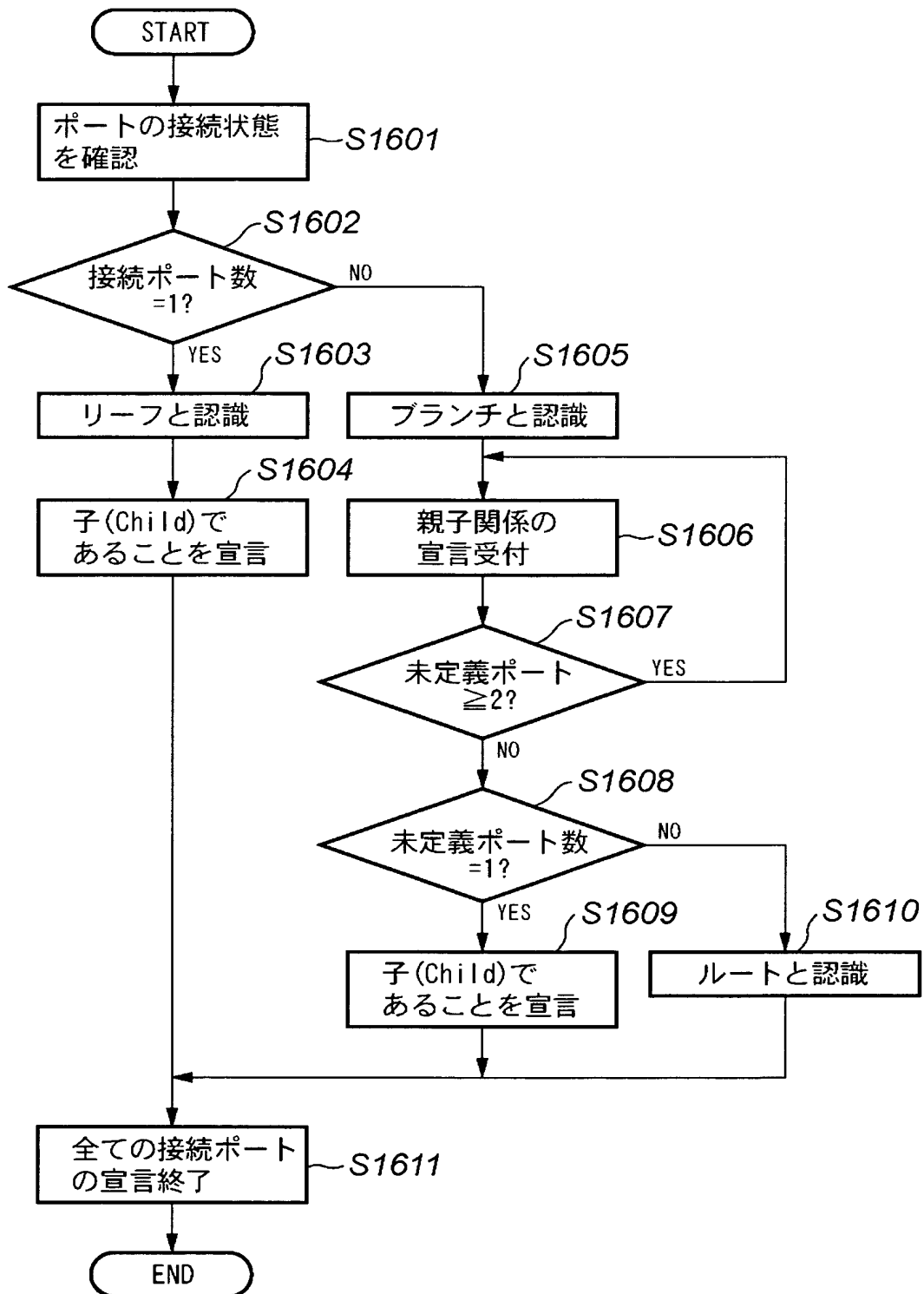




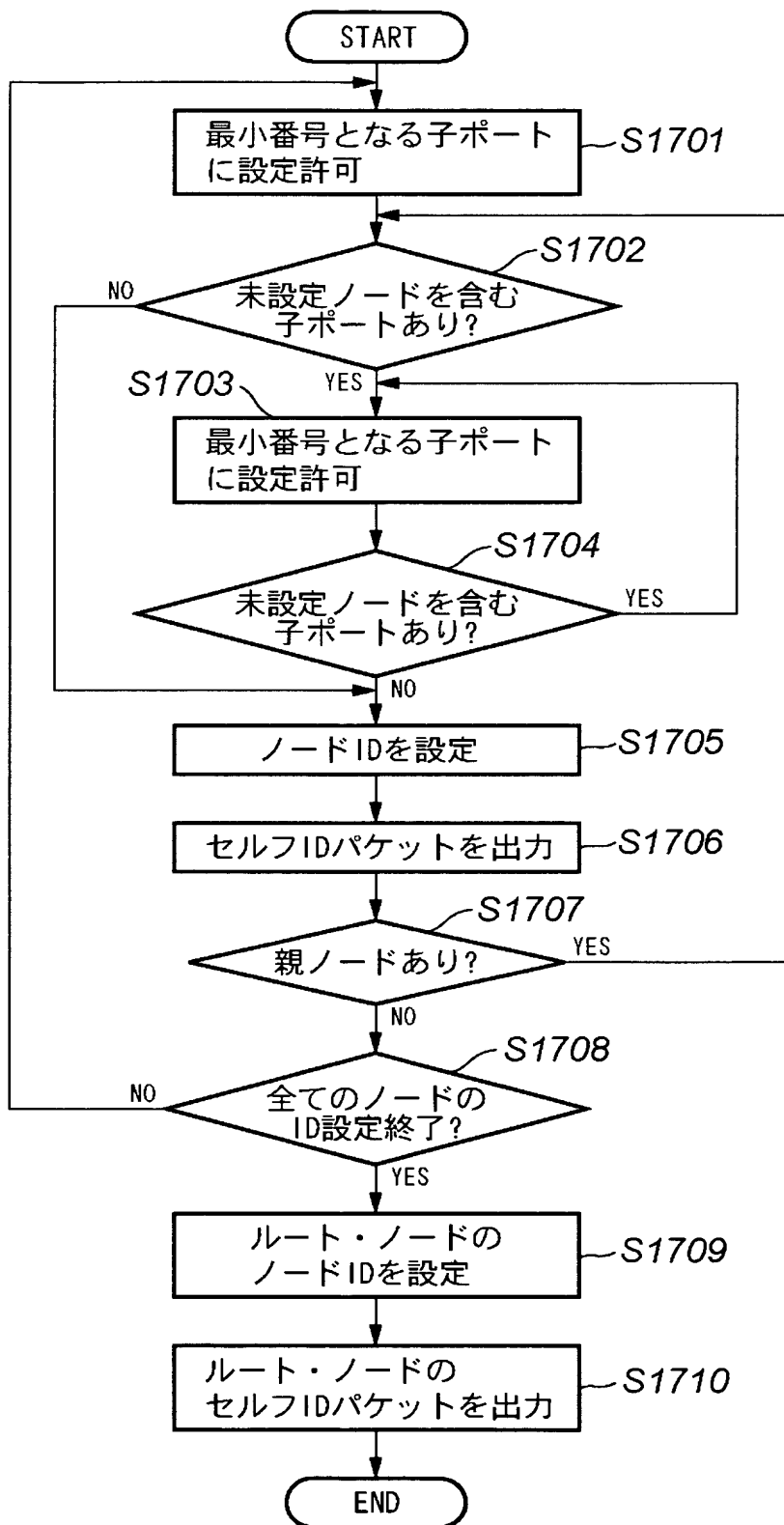
【図 15】



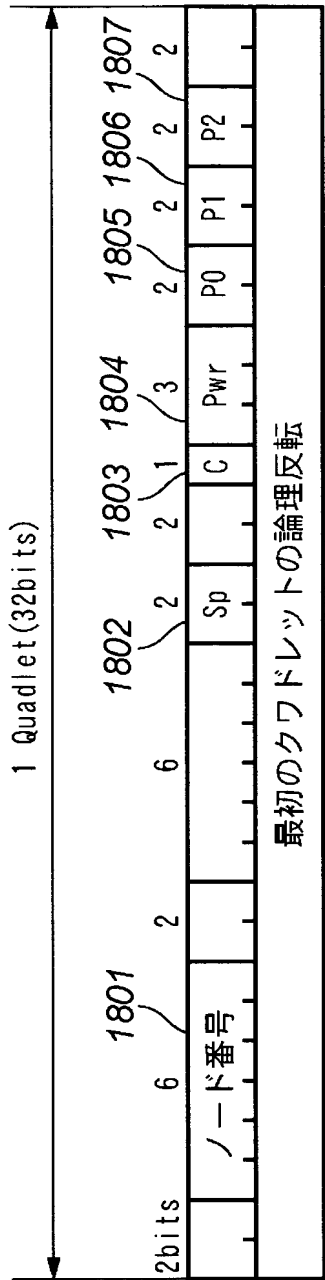
【図 16】



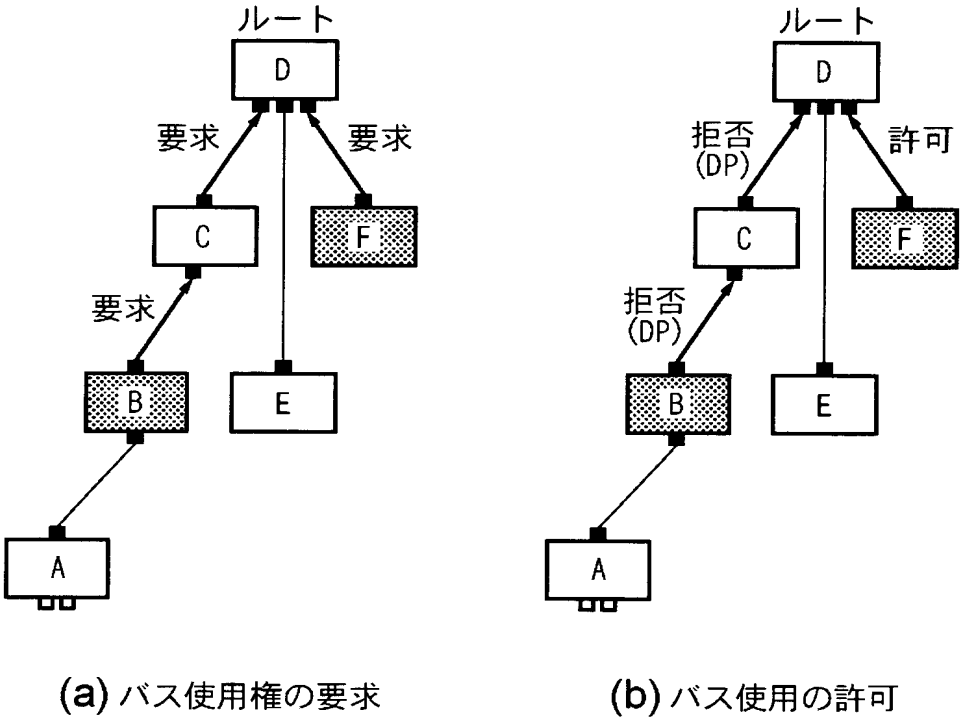
【図 17】



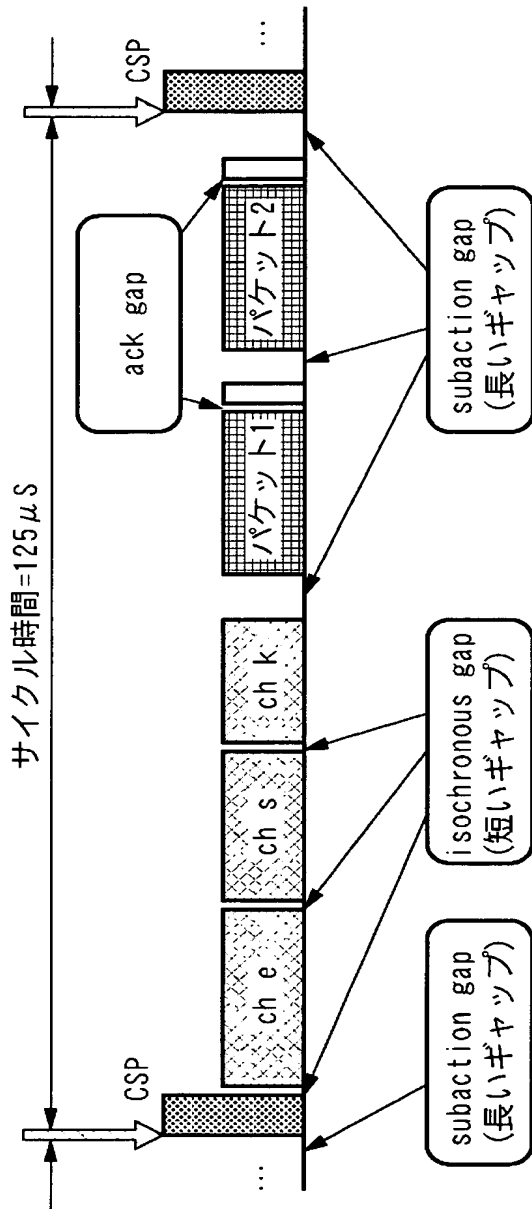
【図 1 8】



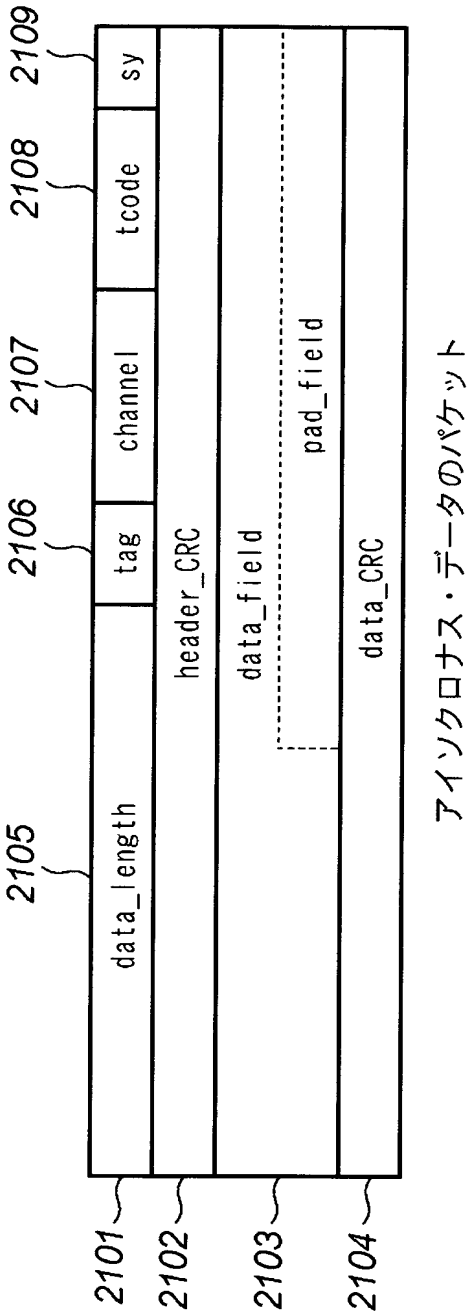
【図 19】



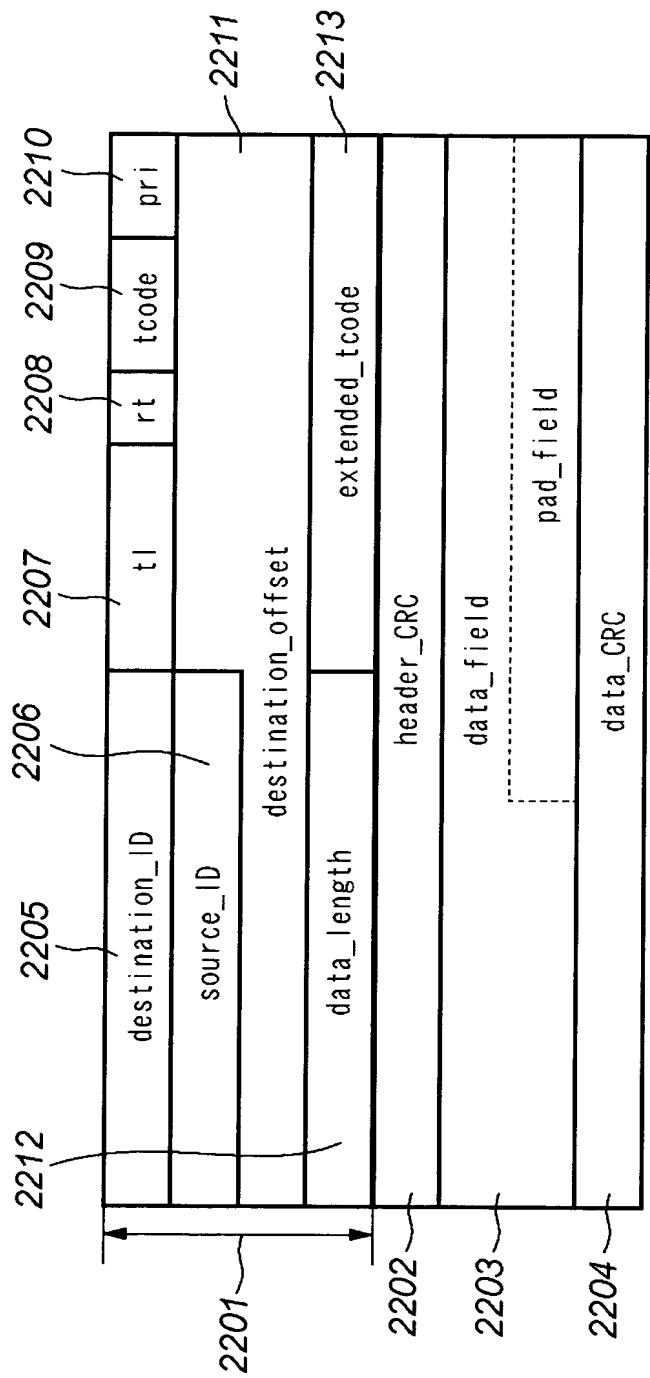
【図 20】



【図 2 1】

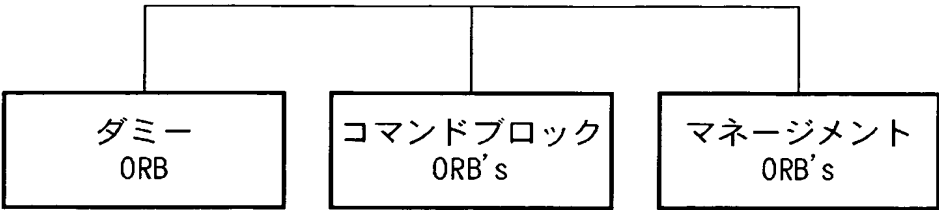


【図 2 2】

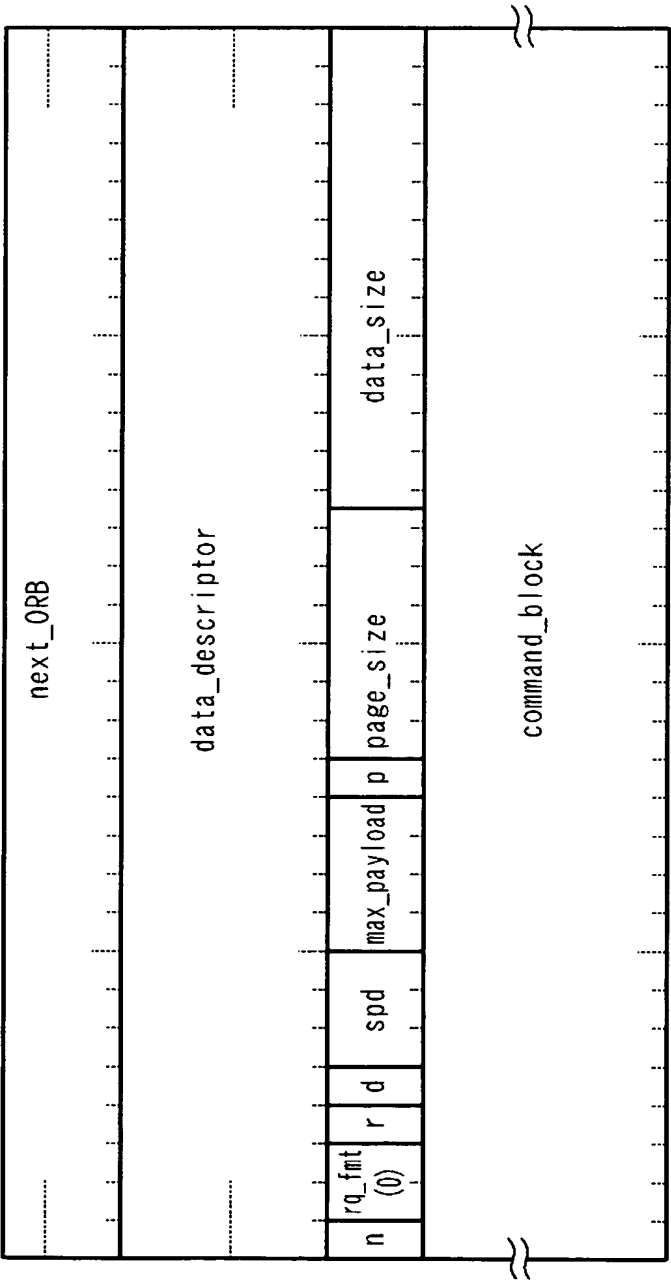




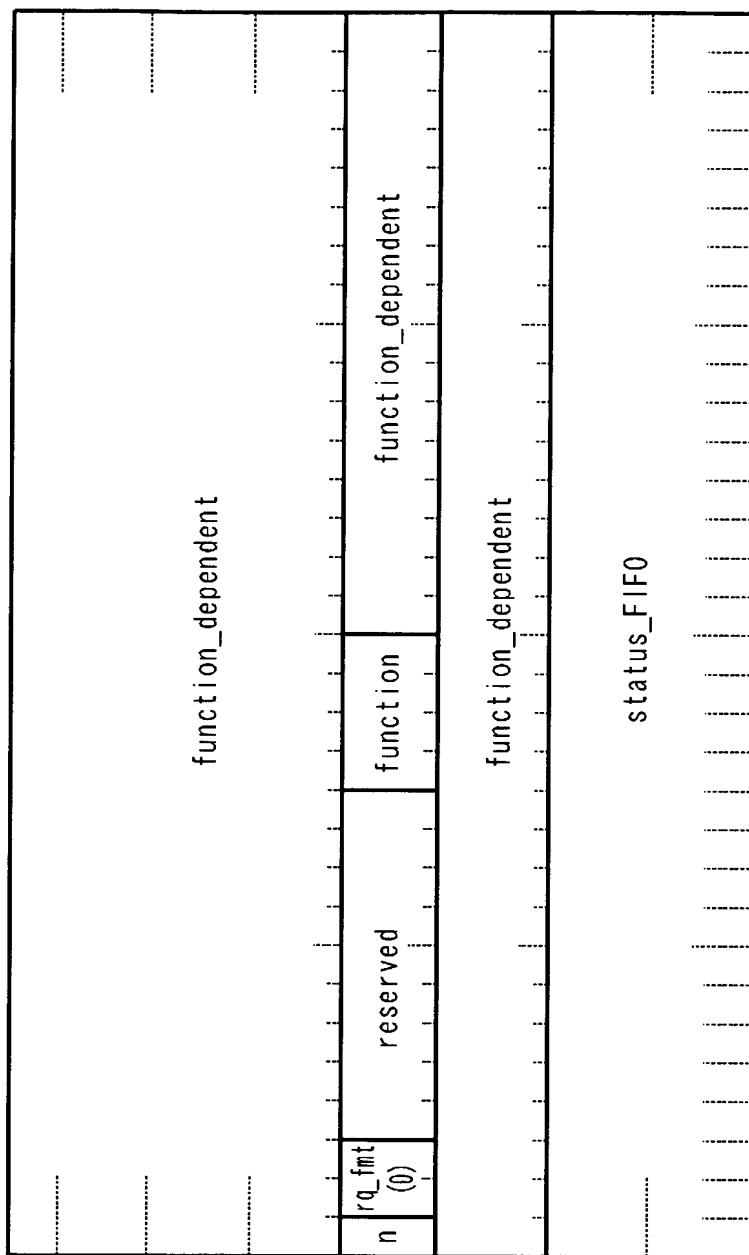
【図 2 3】



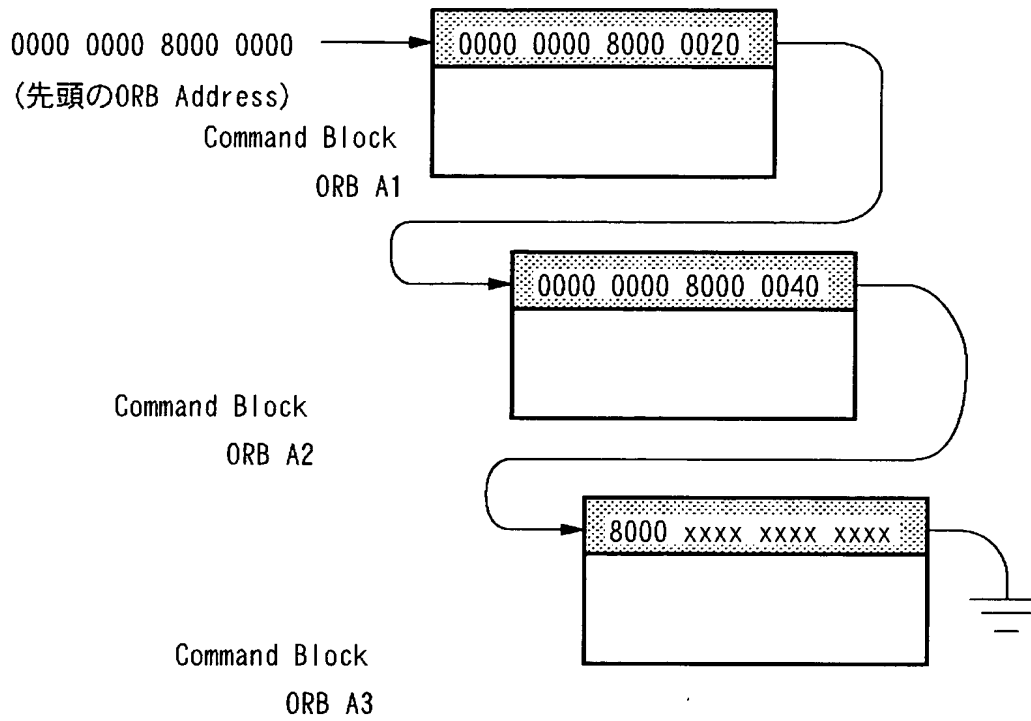
【図 2 4】



【図 25】

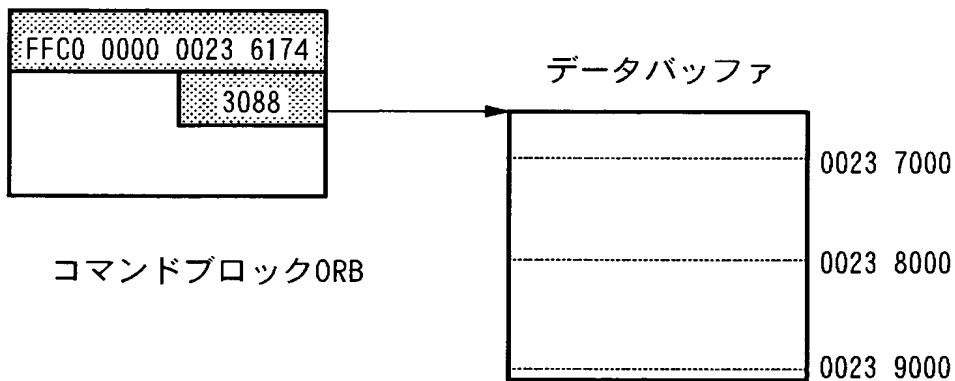


【図 2 6】

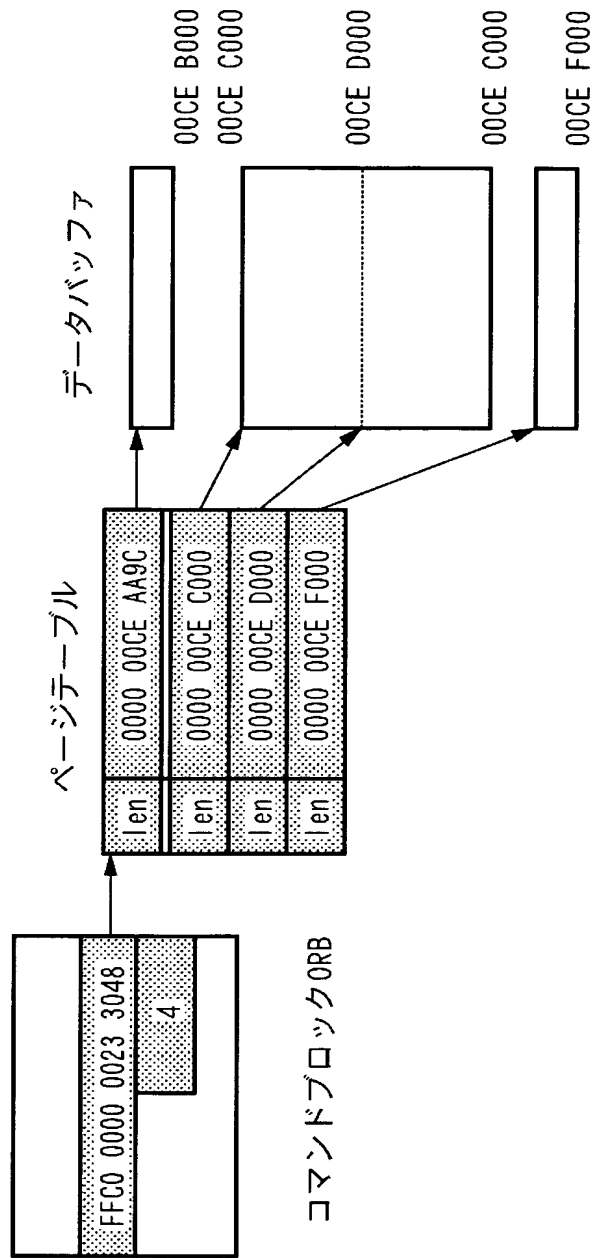


Command Block ORBのLinked List

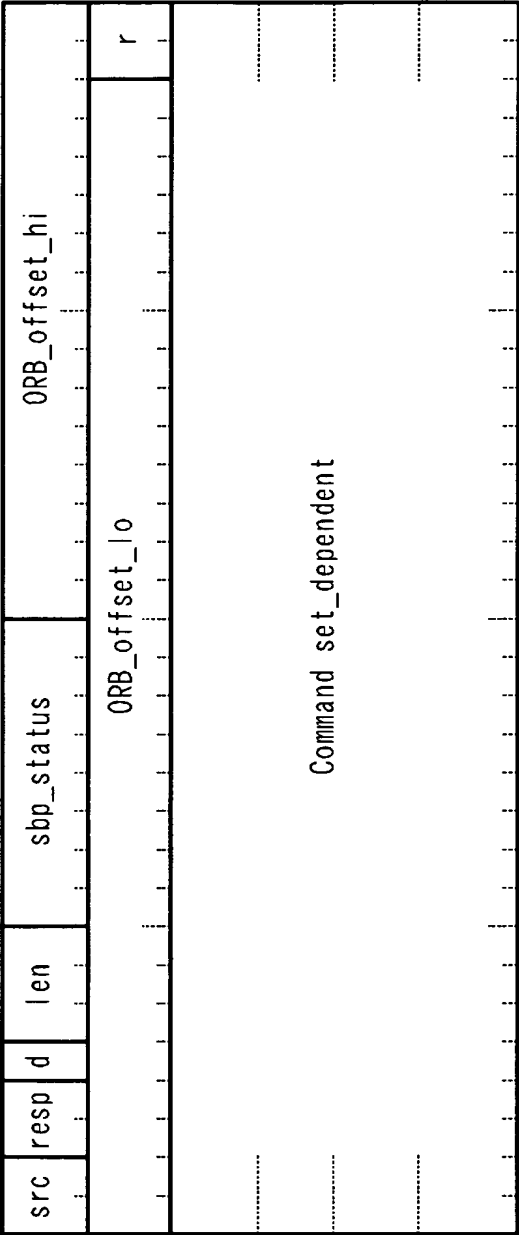
【図 2 7】



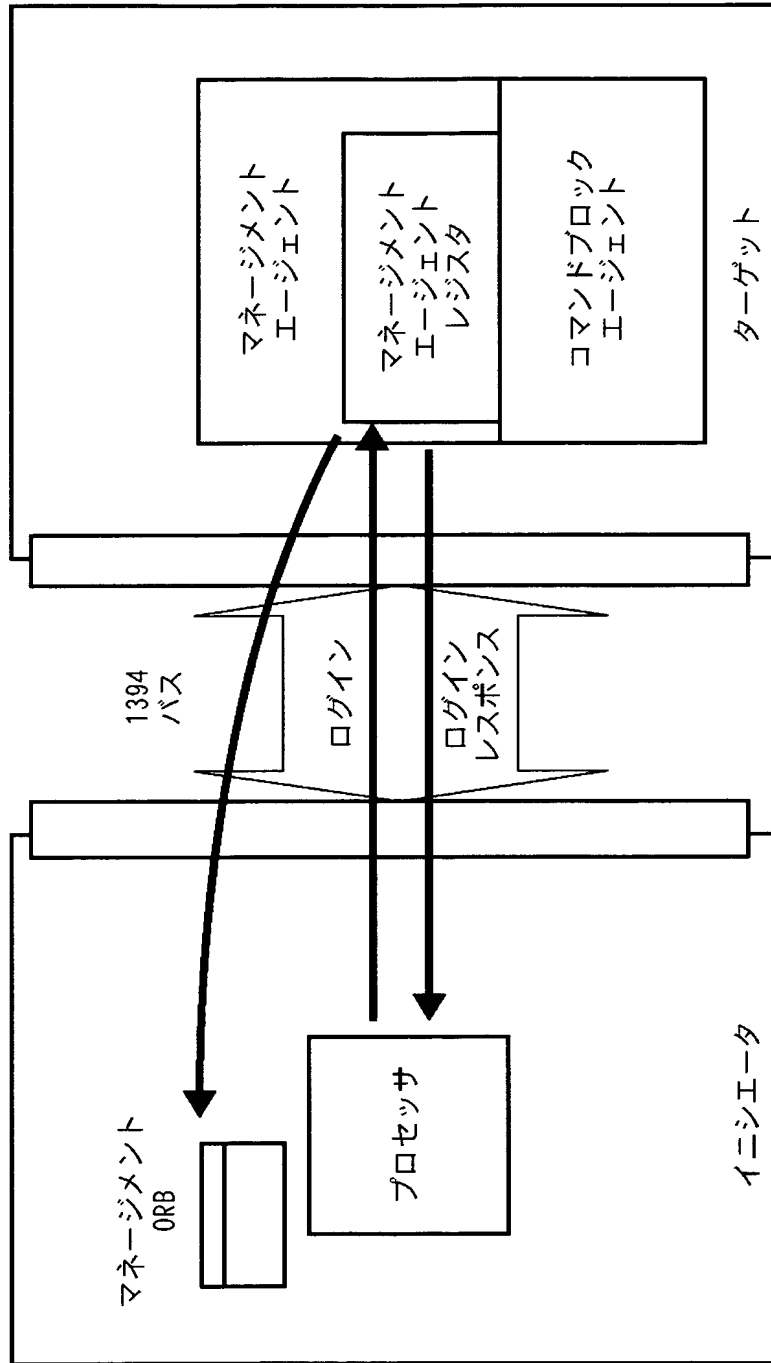
【図 2 8】



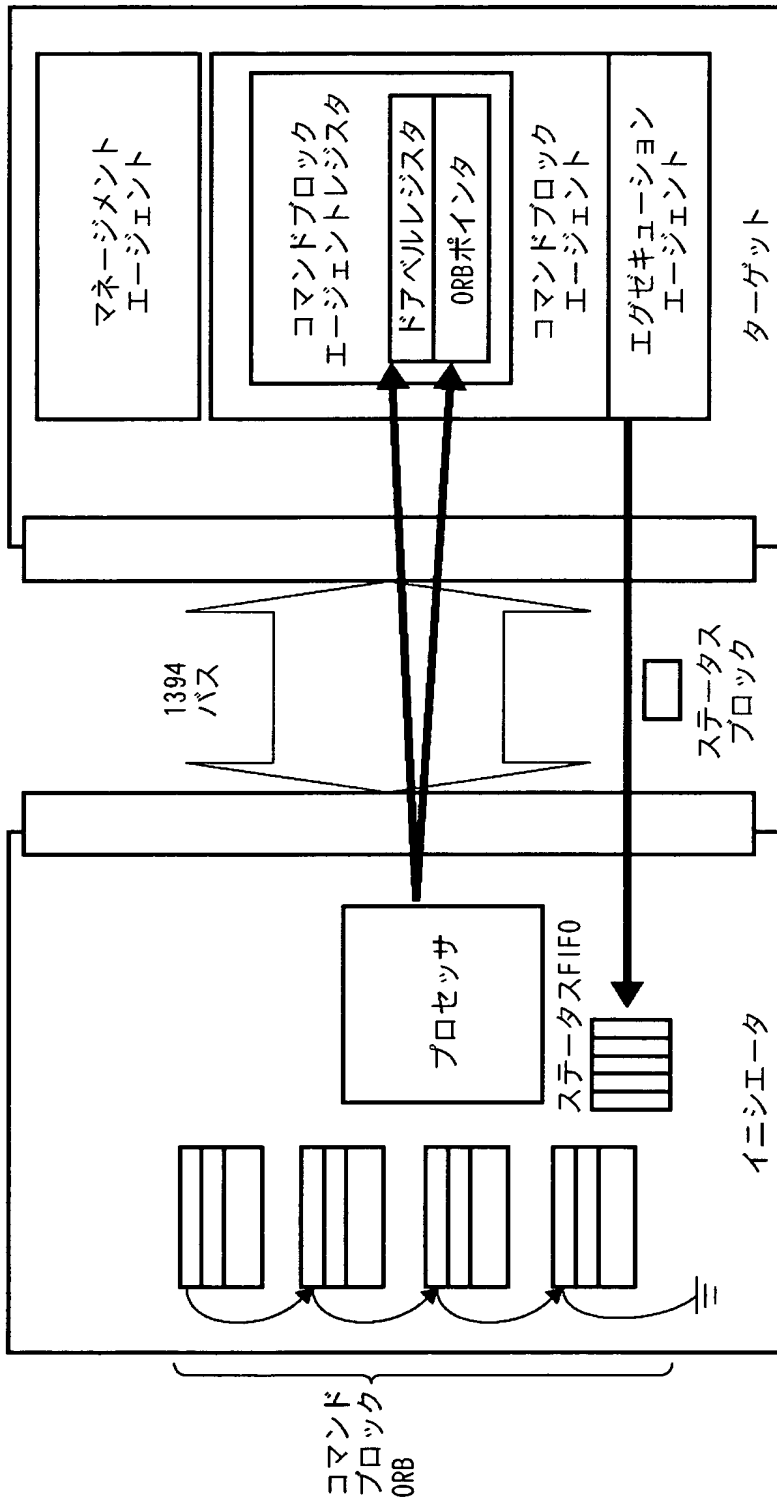
【図 2 9】



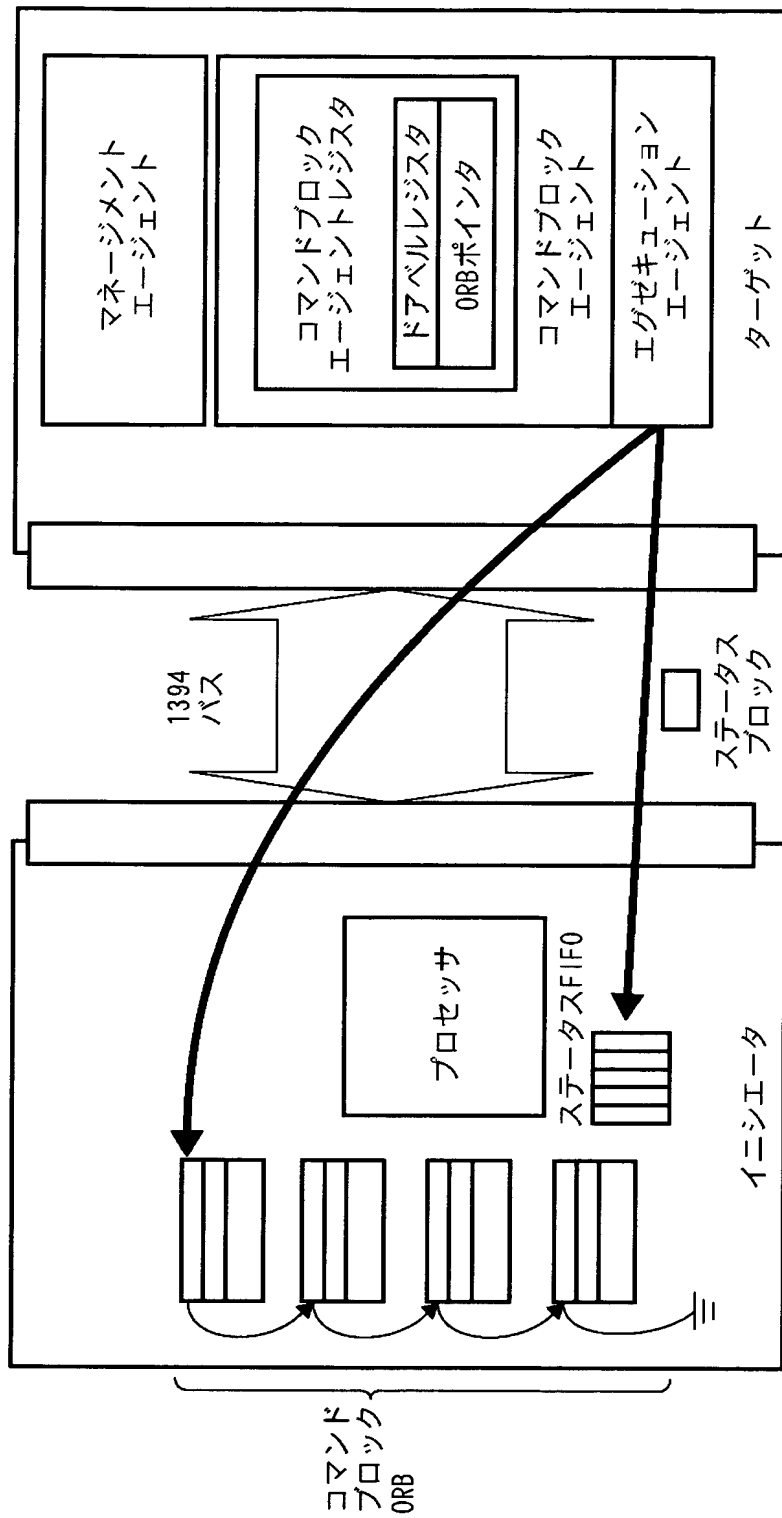
【図 30】



【図 3 1】

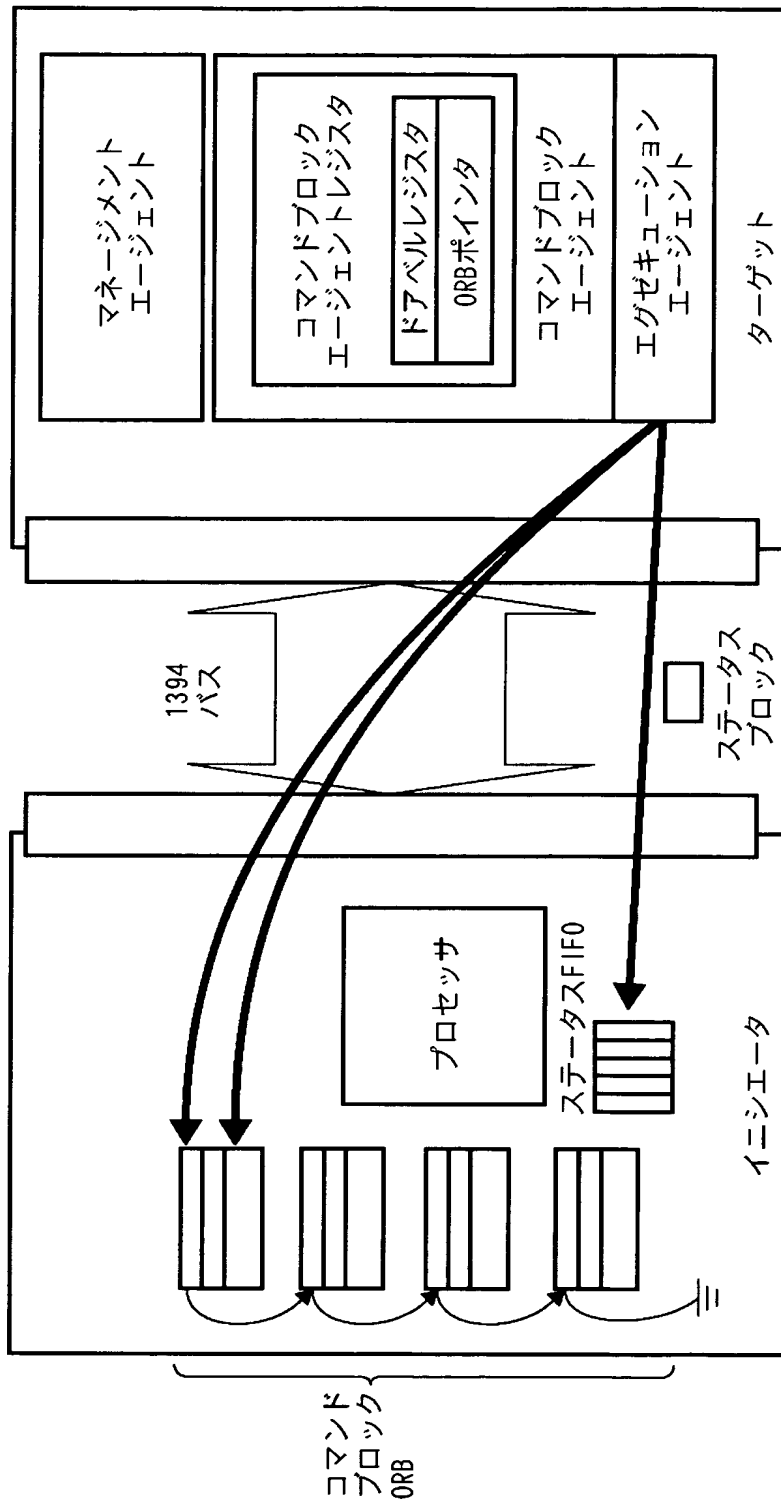


【図 3 2】

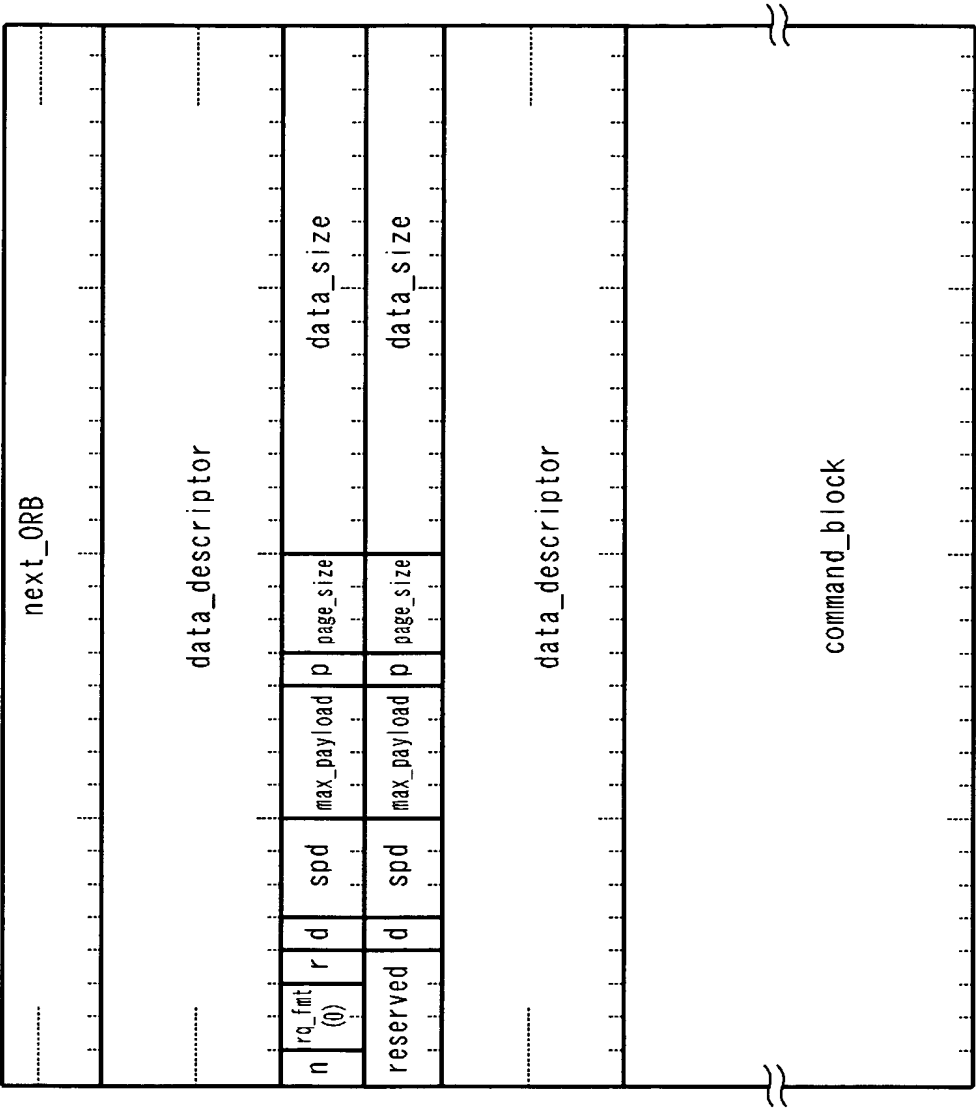




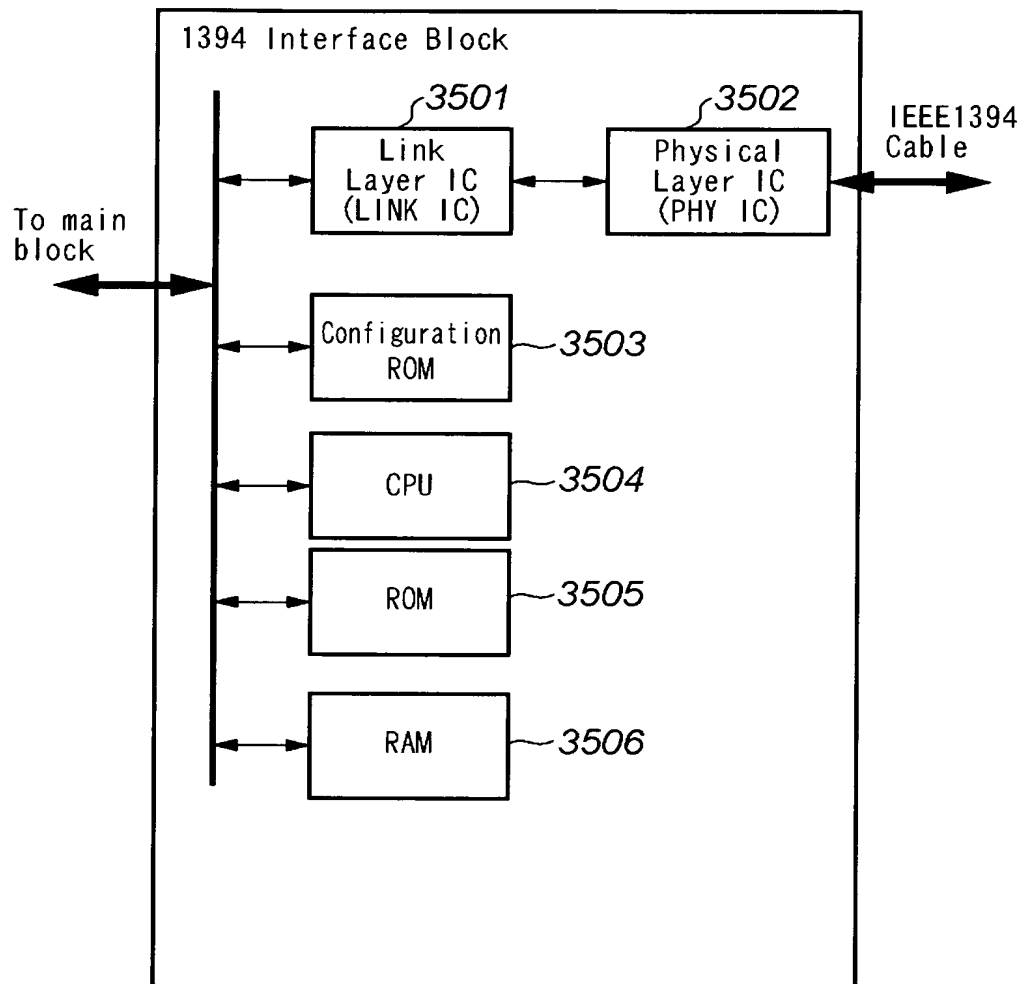
【図 33】



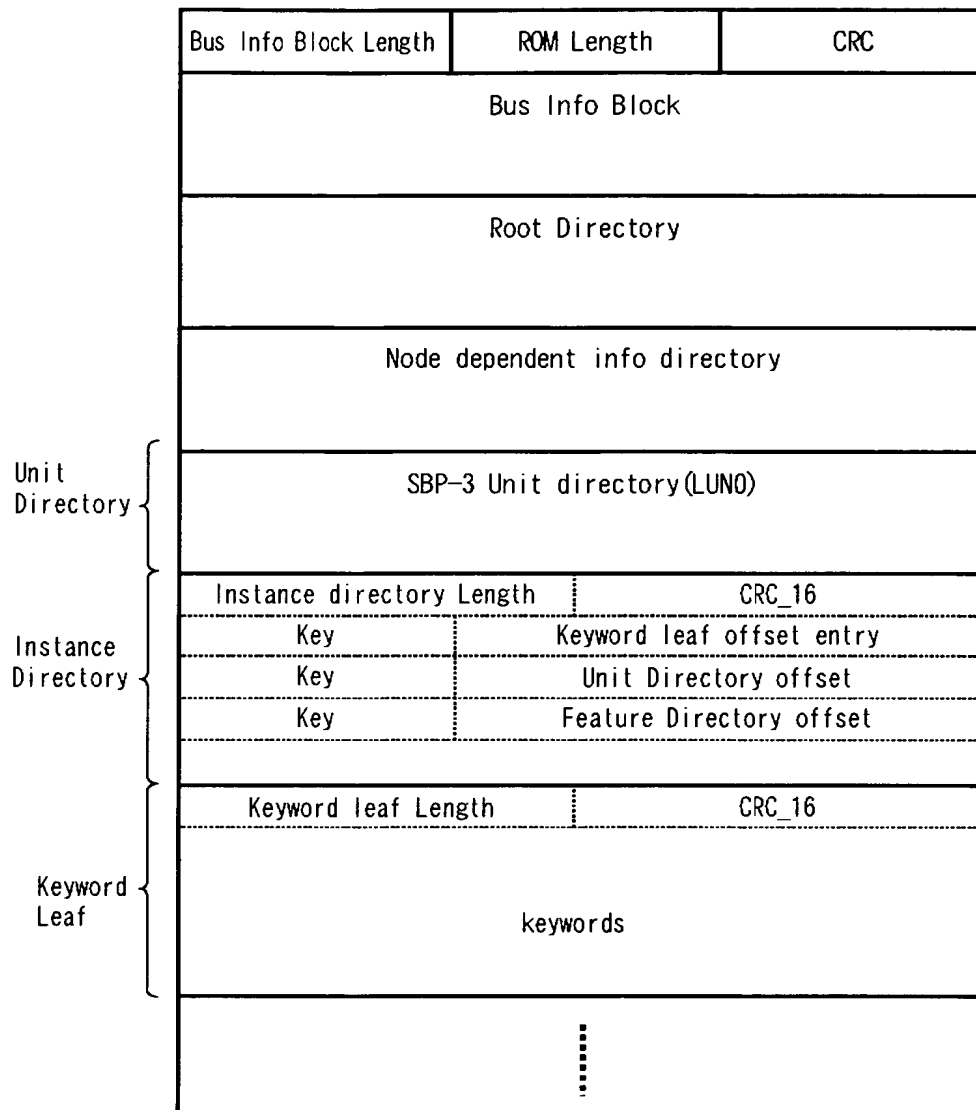
【図 3 4】



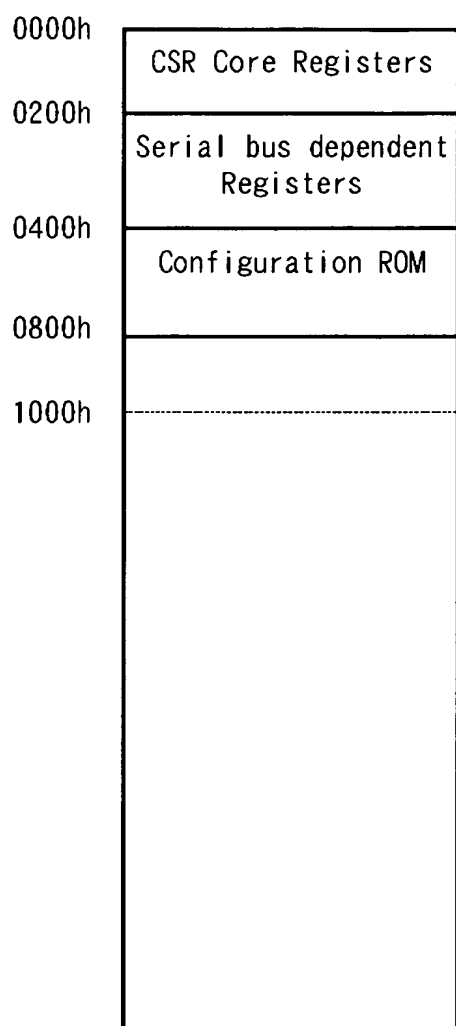
【図 35】



【図 3 6】



【図 3 7】

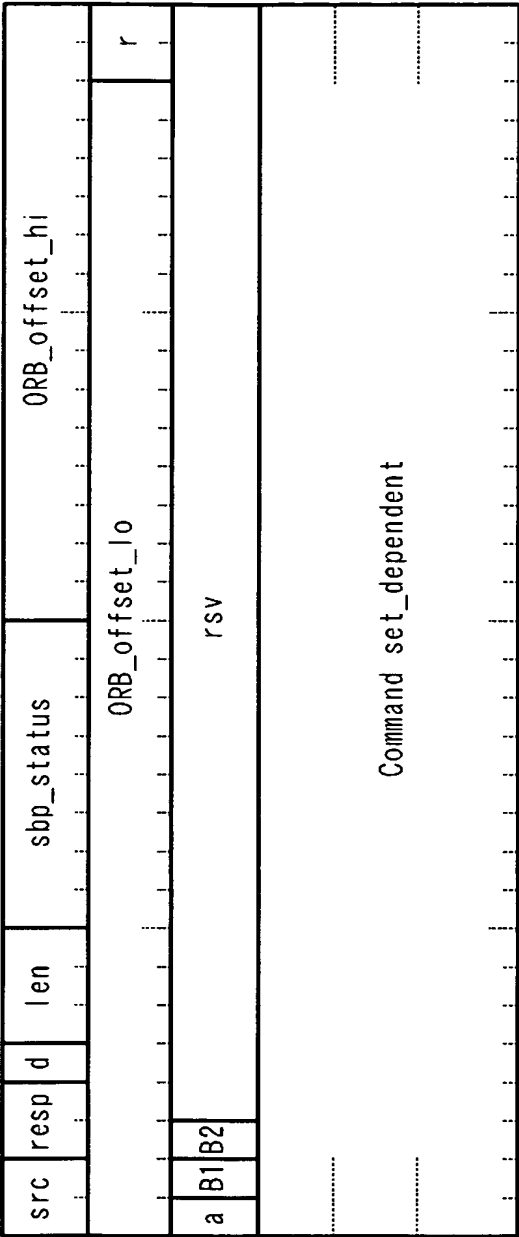


【図 38】

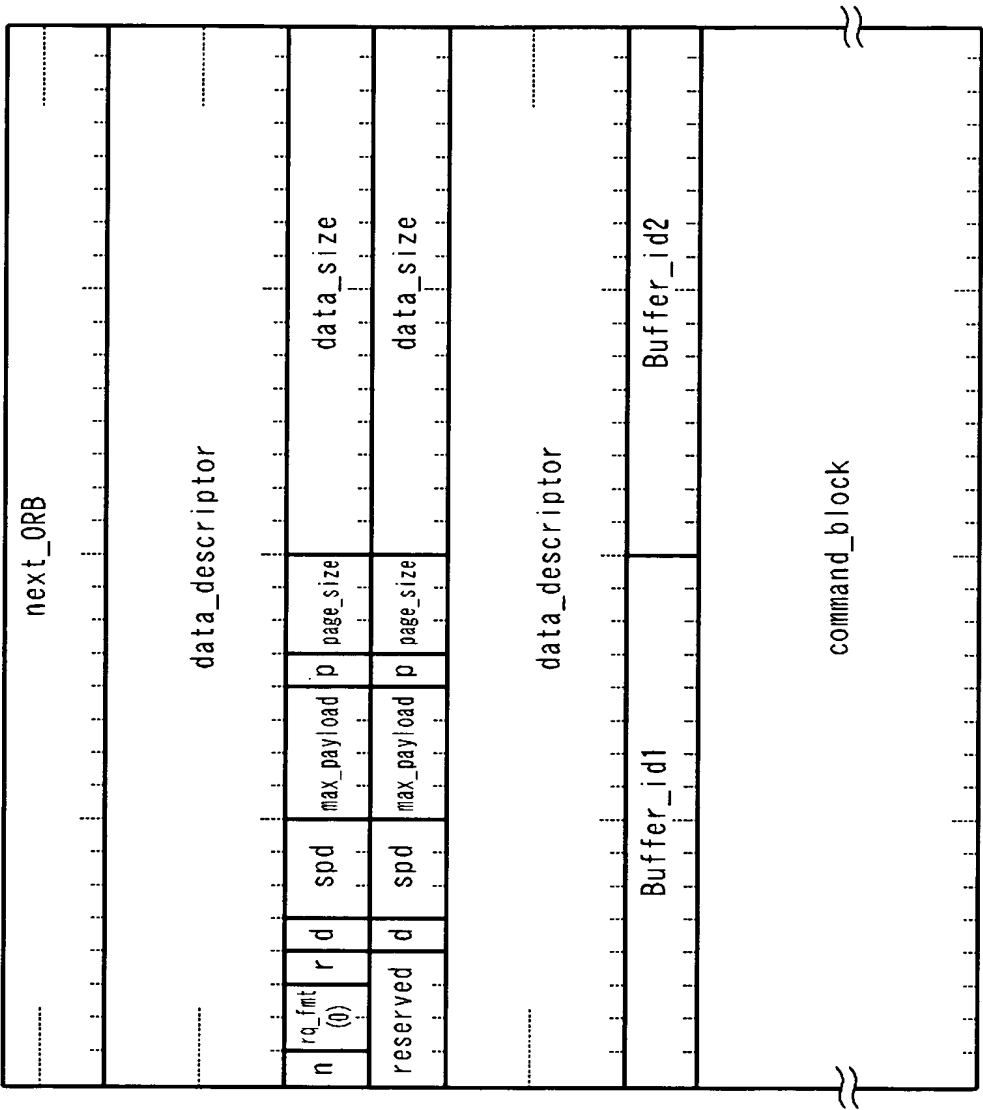
シリアルバスレジスタ

オフセット (16進数)	レジスタ名称	機能
200	CYCLE_TIME	アイソクロナス転送の為のカウンタ
204	BUS_TIME	時間を同期する為のカウンタ
208	POWER_FAIL_IMINENT	電源供給に関するレジスタ
20C	POWER_SOURCE	
210	BUSY_TIMEOUT	トランザクション層の再試行を制御
214~218		予約
21C	BUS_MANAGER_ID	バス・マネージャのノードID
220	BANDWIDTH_AVAILABLE	アイソクロナス転送の帯域を管理
224~228	CHANNELS_AVAILABLE	アイソクロナス転送のチャネル番号を管理
22C	MAINT_CONTROL	診断用レジスタ
230	MAINT_UTILITY	
234~23C		予約

【図 3 9】

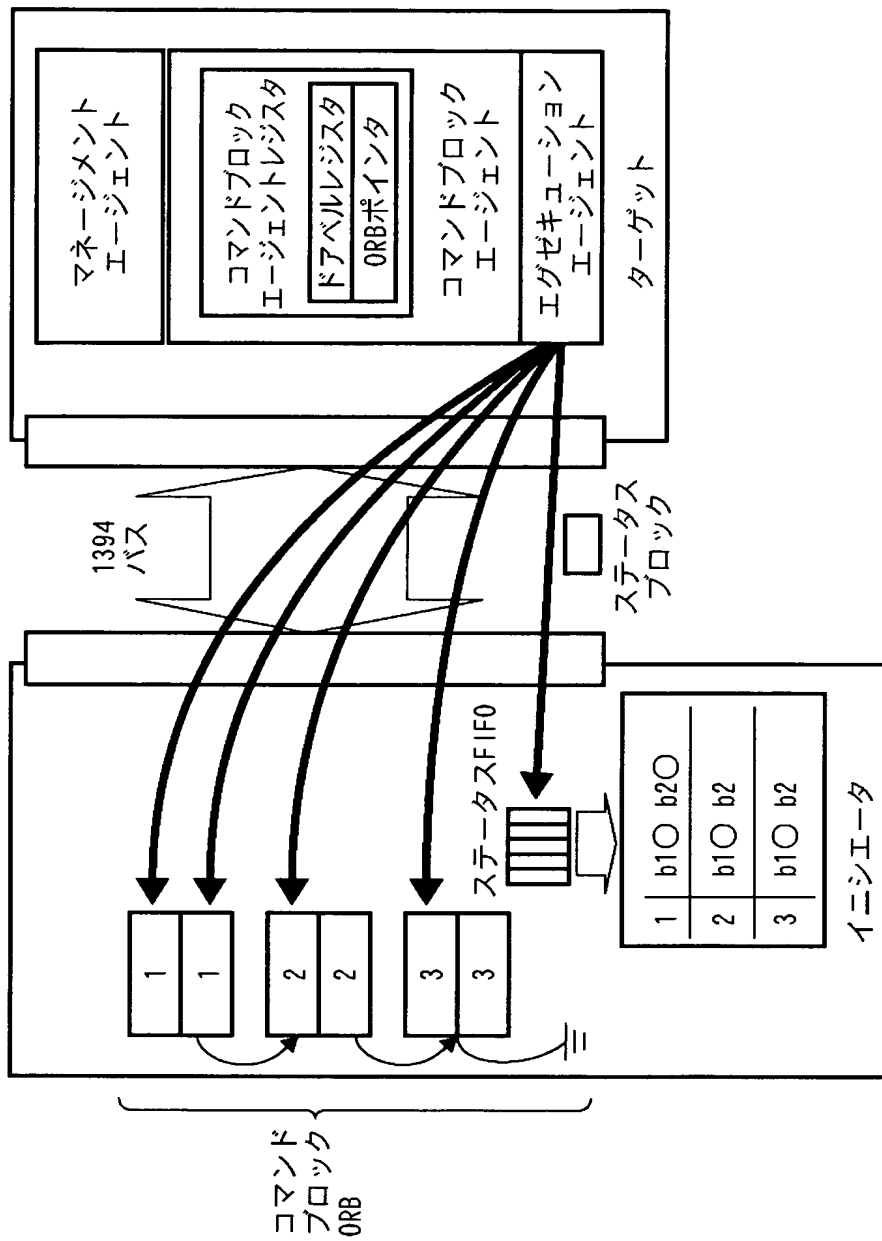


【図 4 0】

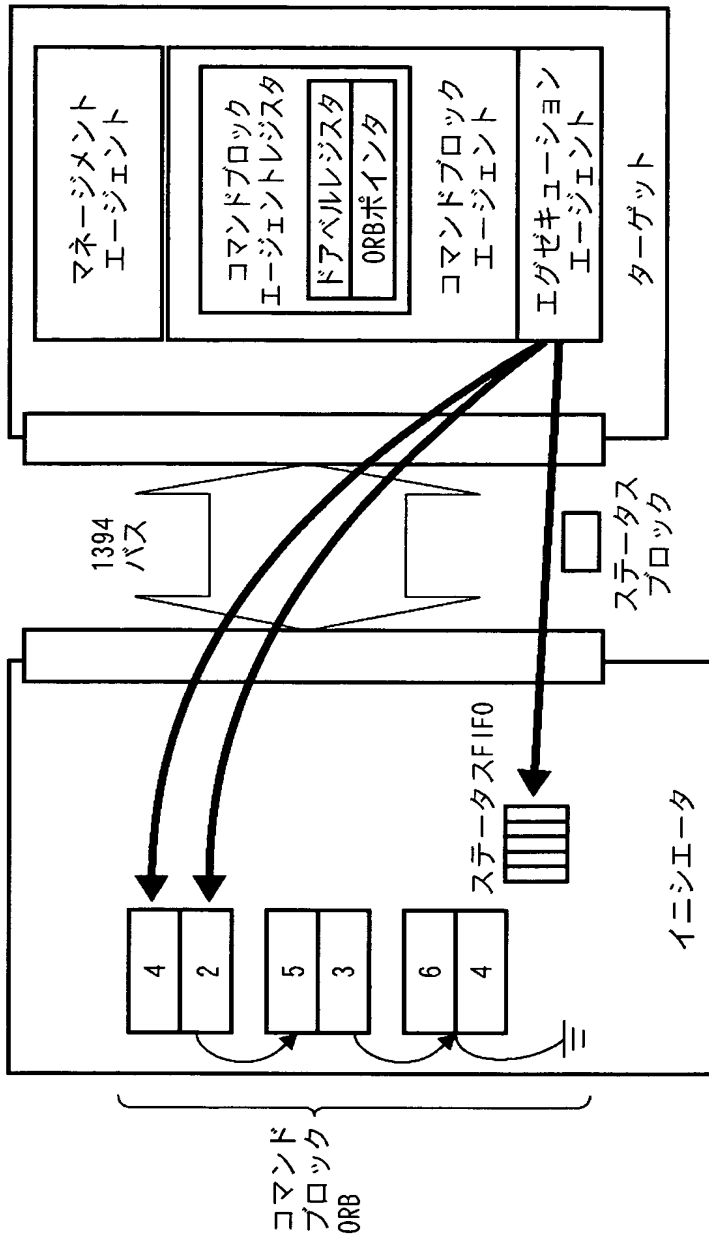




【図 4 1】



【図 4 2】



【書類名】 要約書

【要約】

【課題】 I E E E 1 3 9 4 バスの 2 つのチャネルを用いて効率的なデータ転送を行うこと。

【解決手段】 I E E E 1 3 9 4 に準拠した通信制御バスに接続されたイニシエータとターゲットを含む情報処理システムであって、イニシエータは、ロジカルユニット 1 つを使って、ひとつのコマンド O R B に対し 2 つのデータバッファを用意し、ターゲットは、2 つのデータバッファのいずれか一方に対するデータ通信が完了したことを示すステータスブロックをイニシエータのステータス F I F O に送る。

【選択図】 図 4 1

特願 2 0 0 2 - 2 6 0 3 9 6

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 1 0 0 7 ]

1 . 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都大田区下丸子 3 丁目 3 0 番 2 号

氏 名

キヤノン株式会社